

(51) International Patent Classification 4 : <b>A61B 5/02</b>		A1	(11) International Publication Number: <b>WO 86/ 02250</b>
			(43) International Publication Date: 24 April 1986 (24.04.86)
(21) International Application Number: PCT/US85/01968 (22) International Filing Date: 11 October 1985 (11.10.85)  (31) Priority Application Number: 661,043 (32) Priority Date: 15 October 1984 (15.10.84) (33) Priority Country: US  (60) Parent Application or Grant (63) Related by Continuation US 661,043 (CIP) Filed on 15 October 1984 (15.10.84)  (71)(72) Applicant and Inventor: HASTINGS, Harold, M. [US/US]; 23 Lindbergh Street, Garden City, NY 11530 (US).		(74) Agent: STEMPLE, Dayton, R.; 5200 Empire State Building, New York, NY 10118 (US).  (81) Designated States: AT (European patent), BE (European patent), CH (European patent), DE (European patent), DK, FR (European patent), GB (European patent), IT (European patent), JP, LU (European patent), NL (European patent), SE (European patent), US.  Published With international search report.	
(54) Title: EXTERNAL FETAL HEART MONITOR			
(57) Abstract  A non-invasive, optionally non-ultrasonic method and device for the monitoring of the fetal heartbeat without restricting movement of the mother. The monitoring device and method comprising broadcasting signals from a plurality of abdominally attached microphones (11) by a transmitter (12) carried by the mother to a remote receiver (21) for subsequent processing and filtering by a processor (22), which optionally uses Fourier Transform techniques, from which a resultant electronically enhanced signal may be retrieved from an output device (18).			

***FOR THE PURPOSES OF INFORMATION ONLY***

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	GA	Gabon	MR	Mauritania
AU	Australia	GB	United Kingdom	MW	Malawi
BB	Barbados	HU	Hungary	NL	Netherlands
BE	Belgium	IT	Italy	NO	Norway
BG	Bulgaria	JP	Japan	RO	Romania
BR	Brazil	KP	Democratic People's Republic of Korea	SD	Sudan
CF	Central African Republic	KR	Republic of Korea	SE	Sweden
CG	Congo	LI	Liechtenstein	SN	Senegal
CH	Switzerland	LK	Sri Lanka	SU	Soviet Union
CM	Cameroon	LU	Luxembourg	TD	Chad
DE	Germany, Federal Republic of	MC	Monaco	TG	Togo
DK	Denmark	MG	Madagascar	US	United States of America
FI	Finland	ML	Mali		
FR	France				

## EXTERNAL FETAL HEART MONITOR

This application is a continuation-in-part of Serial No. 06/661,043 filed October 15, 1984. It does not involve federally sponsored research.

## BACKGROUND OF THE INVENTION

This invention relates to new and useful improvements in external fetal heart monitors, and more particularly seeks to provide monitoring of electronically enhanced and processed signals (optionally by Fast Fourier techniques), preferably broadcast from single or multiple, attached, abdominal microphones. Placental blood flow data and abdominal fetal electroencephalographic data may be similarly detected and processed. Other signals such as contraction activity and maternal electrocardiographic data may be simultaneously and similarly processed and/or broadcast.

## FIELD OF THE INVENTION

The primary field of this invention is medicine, as it relates to embryology and obstetrics, and particularly fetal heartbeat monitoring, primarily during labor but possibly at any time after heartbeat detection. In addition, the secondary fields of electronic transmission, electronic enhancing, processing, and filtering e.g. using Fourier Transform techniques, and computers are blended with medicine.

## PRIOR ART

Devices have been used over a long period to monitor fetal heartbeat as an indicator of possible fetal distress, as noted in an early patent 2,536,527 for an external fetal

monitor. Patents 3,280,817 and 3,703,168 use analog filters on fetal heart signals. Patent 3,187,098 separates fetal and maternal heart signals according to their predominant frequency ranges. Patent 3,581,735 shows a method for error detection and correction. Patent 3,762,397 discloses demodulation of cardiovascular blood flow sounds and averaging of synchronized signals.

Patent 3,358,289 shows two signals from two microphones which were combined in the time domain. Patent 3,367,323 discloses the use of multiple electrodes for detecting the abdominal fetal electrocardiogram. The resulting signals are combined by a resistor network. Patent 3,409,737 discloses the use of several microphones on a single elastic belt, where the fetal signal is derived by analog processing of only one switch-selected microphone signal. Patent 4,299,234 discloses a method for the automated combination of two ultrasound and abdominal electrocardiographic fetal heart signals. Other processing techniques include moving average techniques (4,190,866), and the use of both shift registers and autocorrelation (4,037,151). More recently F. Boehm and L. Fields, in Second-generation EFM Is Waiting In The Wings, Contemporary Ob/Gyn, March 1984, p 179, disclose Doppler ultrasound monitors using autocorrelation techniques.

D. Talbert and D. Southall, in Lancet, February 25, 1984, p 426, and April 21, page 907, disclose a broad-band transducer for detecting fetal heart sounds.

Patents 4,281,241, 4,356,475, and 4,356,486 show telemetry in surgical applications. N. L. Barr, in The Radio Transmission of Physiological Information, The Military Surgeon, vol. 114, no. 2, Feb. 1954, p. 79 discloses telemetry of physiological data.

A. M. Flynn, J. Kelly, G. Hollins, and P. F. Lynch, in Ambulation in Labor, British Medical Journal, August 24,

1978, p 591 disclose local radio broadcasting of signals obtained from internal fetal heart monitoring. M. R. Neuman and E. O'Connor, in A Two-Channel Radiotelemetry System for Clinical Fetal Monitoring, Biotelemetry Patient Monitoring 7, 1980, p 104 disclose radiotelemetry of multiplexed uterine pressure and internal fetal electrocardiogram data. Note however that R. W. Saunders, in Fetal Monitoring During Labor, British Medical Journal, December 3, 1983, p 1649, notes the need for satisfactory, practical fetal monitors with telemetry.

Patent 4,378,022 discloses the use of the Fast Fourier Transform in processing a single signal for in vivo analysis of heart valve prostheses. L. R. Rabiner and R. W. Schafer, Recursive and Nonrecursive Realizations of Digital Filters Designed by Frequency Sampling Techniques, IEEE Transactions on Audio and Electroacoustics, vol. AU-19, no. 3, Sept. 1971, p. 200, disclose the use of Fourier Transform and frequency domain filtering to emulate certain analogue filters applied to single signals. The Fast Fourier Transform is a well-known algorithm with many implementations, cf. A. Aho, J. Hopcroft and J. Ullman, Design and Analysis of Computer Algorithms, Addison-Wesley, 1974, and E. Horowitz and S. Sahni, Fundamentals of Computer Algorithms, Computer Science Press, 1978. C. S. Burrus and T. W. Parks, in DFT/FFT and Convolution Algorithms, John Wiley and Sons, 1985 disclose improvements to the Fast Fourier Transform of a real signal. C. S. Burrus and T. W. Parks, in DFT/FFT and Convolution Algorithms, John Wiley and Sons, 1985 disclose improvements to the Fast Fourier Transform of a real signal. However, no such algorithm is disclosed for the inverse Fast Fourier Transform. None of the above prior art or art known to the applicant, except patent 4,378,022, shows the Fast Fourier Transform applied to time-domain heart signals.

J. L. Hughes and C. A. Phillips, in On-Line Digital Processing of Uterine Contraction Waveforms, Proc. IEEE 1977

Nat. Aerospace and Electronics Conference NAECON '77, Dayton, Ohio, 17-19 May 1977, p. 244, disclose digital processing of uterine and fetal heart rate data.

The design of amplifiers and filters using operational amplifiers (op-amps) is well-known, e.g., W. G. Jung, IC Op-Amp Cookbook, 2nd ed, Howard W. Sams, 1983, and C. R. Wyss, Planning a Computerized Measurement System, BYTE, April 1984, p 114.

Previous fetal monitors have used four main techniques, namely (1) external detection of fetal heart sounds (fetal "phonocardiograms"), (2) external ultrasound and Doppler ultrasound, (3) external (abdominal) electrocardiogram (fetal ECG), and (4) internal fetal ECG. Previous data processing techniques included active audio filters, moving average analysis, and autocorrelation analysis. All of these techniques have disadvantages.

In some cases external phonocardiographic or electrocardiographic (ECG) monitors have failed to obtain a usable signal, thus requiring invasive internal monitors or ultrasound. External fetal ECG signals are disrupted by significant electrical activity of the uterus and maternal heart. External phonocardiographic monitors may be similarly disrupted by sounds.

In cases where observations of the fetal heartrate suggests a risk to the fetus, it becomes important to monitor the variability of short-term fluctuations in the rate. In general variability cannot be accurately determined from existing direct external monitors and non-Doppler ultrasound monitors. In particular, both classes of monitors overestimate variability. This causes uncertainty when lower than expected variability is detected, and frequently leads to unnecessary Caesarian sections.

BAD ORIGINAL

SUBSTITUTE SHEET

The use of internal monitors to avoid these problems requires rupturing the membranes, carries the risk of infection, may alter the course of labor and introduces an added element of risk into the birth canal.

Recent Doppler ultrasound monitors using autocorrelation as a filtering technique have also been designed to overcome these problems, and provide a useful indication of variability. However, the use of any ultrasound during labor poses risks to the fetus which may absorb significant amounts of energy, especially when ultrasound monitoring is used during a significant part of a long labor.

In addition, all previous monitors severely restrict mobility during labor, both because fetal and maternal movement is restricted in order to obtain and maintain an adequate signal, and because of wires connected to the mother and external apparatus. This restriction is unpleasant, makes it difficult to determine the effects of maternal movement frequently suggested to improve the state of the fetus, and may extend the duration of labor. The prolongation of labor again places additional direct burdens upon the fetus and mother, as well as indirect burdens if it necessitates medication and procedures not otherwise required.

#### SUMMARY OF THE INVENTION

The invention consists of a fetal monitor which uses one or preferably multiple microphones to detect fetal heart signals. These signals as well as auxiliary data on contractions, maternal heartbeat, etc, are optionally but preferably broadcast locally by a lightweight, low power transmitter worn by the mother which allows full mobility during labor monitoring.

BAD ORIGINAL

-6-

The signals are sent directly or by broadcasting to a processing unit where the signals are rapidly filtered and combined, using primarily digital techniques to produce a filtered and enhanced signal of fetal heartbeat, in synchrony with slightly delayed auxiliary data. In addition these techniques produce a direct measure of variability in terms of the range of heart rates present in each sampling cycle. Both the quality of the enhanced signal and the additional measure of variability decrease current use of internal monitors, ultrasound, and Caesarian sections.

The instant monitor overcomes these difficulties in several ways. First, the use of multiple microphones or other sensors, together with suitable processing techniques to obtain a single combined, enhanced signal, allows for better monitoring during maternal or fetal movement. Mobility is further facilitated by the use of local transmission in place of wires. In essentially all circumstances, these new processing techniques yield accurate data on fetal heartrate and its variability from direct external monitoring without ultrasound or invasion.

The signal processing is done by a fast microprocessor and associated circuitry using the Fast Fourier Transform (FFT), filtering techniques in the frequency domain together with an appropriately chosen sampling rate, and the inverse FFT. Each fetal heart signal and a maternal heart signal is amplified (then optionally partially demodulated), filtered to permit accurate sampling, and converted to a digital signal at a sampling rate chosen to completely reject 60 Hz electrical noise. The FFT is used to compute the frequencies and respective amplitudes present in each heart signal over the most recent sampling period. A subsequent linear or nonlinear, digital filter is applied to emphasize desired frequencies. In particular, noise due to the maternal heartbeat may be efficiently filtered out from

BAD ORIGINAL

SUBSTITUTE SHEET



the fetal heart signals. The power spectrum (energy at each frequency) of each fetal signal is computed. This data is used to combine frequency data from the multiple fetal signals in proportion to their signal quality so as to obtain a single enhanced, filtered signal in the frequency domain. The composite power spectrum is then computed to provide data on the range of fetal heart rates. The enhanced frequency signal is transformed by the inverse FFT in order to obtain an enhanced, filtered, relatively noise-free fetal heart signal.

This enhanced signal lags only slightly behind the original signals. The auxiliary inputs may be presented in real time to signal events such as the onset of contractions, or similarly delayed (and processed if desired) to be in synchrony with the enhanced fetal heart signal. The enhanced, filtered fetal heart signal may be further processed using moving average or other techniques to obtain information on instantaneous heart rates, and provides better data on heart rates, variability, acceleration and deceleration than provided by the original fetal heart signals.

Note in particular that the transform techniques plus filters in the frequency domain combine the performance of active audio filters, sharp notch filtering of 60 Hz noise, and further noise reduction due to implicit time averaging in the frequency domain and an optimal combination of separate signals. This noise includes the maternal heartbeat, effects of uterine contractions, and other noise. In addition, the multiple microphones mean that fetal movement will not prevent obtaining a usable signal of the fetal heartbeat.

Finally, the frequency data obtained by the FFT and filters is itself displayed as another indicator of short-term variability. In addition to the improved

measurement of peak-to-peak variability due to better signal quality, Fourier transform techniques offer an additional measure of variability.

Furthermore, since the amplitude of the fetal heartbeat depends in part upon the strength of the beat, among other factors, the enhanced signal obtained from the instant monitor allows measurements of relative changes in the strength of the fetal heartbeat in addition to the fetal heart rate.

The instant fetal monitor thus reduces exposure to internal monitors, ultrasound, and Caesarian sections through more accurate determination of fetal conditions, while enhancing comfort and mobility, which reduce the difficulty and duration of labor.

Moreover, the use of multiple signals combined in the frequency domain allows monitoring the abdominal fetal electrocardiogram as well as blood flow in the placenta, in which each major vessel does not move during labor, but offers only a weak signal.

This method can also be used to detect and enhance other signals in a large variety of medical applications, such as fetal electroencephalography.

Further to this summary of the invention, the specific nature of which will be more apparent, the invention will be more fully understood by reference to the drawings, the accompanying detailed description and the appended claims.

#### BRIEF DRAWING DESCRIPTION

Fig. 1 is a perspective view of a pregnant female,

with attached microphones, transmitter unit, and antenna broadcasting to processing and monitor units in accordance with this invention;

Fig. 2 is a more detailed diagrammatic view of the apparatus of Fig. 1;

Fig. 3 is a detailed diagrammatic view of a microphone unit of Fig 1;

Fig. 4 is an electronic circuit diagram for the amplifier of Fig. 3;

Fig. 5 is a detailed electronic circuit diagram of subsequent amplifier stages contained in the transmitter, receiver, and processor of Fig. 1, including filtering and demodulating components;

Fig. 6 is a detailed diagrammatic view of the transmitter unit of Fig. 2, including a multiplexer;

Fig. 7 is a detailed diagrammatic view of the receiver unit of Fig. 2;

Fig. 8 is a detailed diagrammatic view of the demultiplexer of Fig. 7;

Fig. 9 is a detailed diagrammatic view of an alternate embodiment of the demultiplexer of Fig. 7;

Fig. 10 is a detailed diagrammatic view of the major electronic components of the processor unit of Fig. 2;

Fig. 11 is a detailed diagrammatic view of the flow of information in the processor unit shown in Fig. 2;

Fig. 12 is a detailed diagrammatic view of or-

BAD ORIGINAL

dual-ported random access memory (RAM) from the processor unit of Fig. 10;

Fig.13 is a partial detailed diagrammatic view of another dual-ported RAM from the processor unit of Fig. 10; and

Fig. 14 is a graphic view of the representation of two different fetal heart signals, showing differences in variability.

#### DESCRIPTION OF THE PREFERRED EMBODIMENTS

##### Overview

The main part of the present fetal monitor consists of the electronic (analog and digital) processing devices, as described below. These devices allow the practical use of external monitoring such as the use of one or more fetal microphones as required, and also the use of local broadcasting to facilitate maternal mobility.

Referring to Fig. 1, multiple microphones 11, are taped to a woman's abdomen. The number of microphones can be varied depending upon circumstances affecting the level of the signal and amount of noise. Such circumstances might include position and depth of the fetus, location of the placenta, amount of external noise, and other factors. The signal-to-noise ratio of the overall fetal heart signal (before or after processing) depends inversely upon the number of said microphones. Several microphones may be combined into a single unit if desired in order to minimize the number of units worn by the mother. Between one and eight microphones may be used, three to six is most

BAD ORIGINAL

desirable, and four is preferred. Added microphones give a potentially better signal but require transmission of more signals. Microphone denotes a suitable low frequency pressure transducer for detecting the fetal heartbeat or sounds in the placenta, e.g., suitably designed acoustic microphones, photoplethysmographic detectors, etc.

The positions of these microphones 11 are selected so as to maximize detection of the fetal heart signal during expected maternal and fetal movement, and in the presence of expected noise. This may be done by monitoring the detected fetal heart signals, e.g., by test jacks on the transmitter or processing unit or both. Alternatively, the microphones may be placed in standard positions chosen to allow for movement of the fetus during labor. The microphones may be easily repositioned at any time.

Alternatively, if desired, microphones may be used to detect and monitor placental blood flow, or plural electrodes may be used to detect the abdominal fetal electrocardiograph.

Referring to Figs. 1 and 2, the microphones 11 convert fetal heart beats to electrical signals which are broadcast by a transmitter 12. The transmitter may also broadcast signals from auxiliary devices 15 used to detect contractions, uterine tonus, electromyographic data, intra-uterine pressure, maternal heartbeat, etc. In particular, some noise in the fetal heart signals is due to the maternal heartbeat, which may be detected, processed, and largely eliminated. This requires that a microphone or other detector similar to those used to detect fetal heartbeat be positioned and used to detect the maternal heartbeat. The resulting signal is treated as an auxiliary input up to the processing stage.

The output from the transmitter, containing all of these signals, preferably in a frequency modulated,

ultra-high frequency signal, is broadcast by a flexible wire transmitting antenna 14 contained in a belt 13, which also holds the transmitter, worn by the mother.

The transmitted signal is broadcast to a receiving antenna 16, connected to a receiver 21 and processor 22, the latter enhancing the signal and relaying the enhanced signal to display devices 18.

Referring to Fig. 3, the preferred microphones 11 are modified Tapho micorphones of Talbert and Southall containing an improved amplifier which incorporates an active low-pass audio filter. The microphone 11 includes a transducer 31 and a microphone amplifier (with self-contained filter) mounted on a circuit board 32, housed in a metal or plastic housing 33. The transducer may be a sensitive, low noise, ceramic phono type unit such as the BSR SX6H. The transducer is acoustically coupled to the external abdominal wall by a light plastic button 34 in contact with said wall and a rubber coupling 35 connected to the transducer 31 and in contact with the button 34. The transducer is clamped between rubber supports 36. The housing is lined with acoustic insulation 38 so as to reduce noise. Additional insulation 39 serves as a pad between the housing and the mother's skin. The transducer may be attached by adhesive such as double sided tape 40. Various bolts (501, and not shown), a metal plate 502 strands 505, and a shielded cable 503 connected to a jack 37 complete the housing. The housing 33 may be about 5 cm in diameter and 2.5 cm high, and is designed so that its weight of about 100 gm provides a suitable acoustic reference.

One or more transducers, as described above, may be mounted in a single housing. In addition, a pickup for an external intrauterine pressure sensor may be combined with such a unit to further the number of separate units connected to the mother.

BAD ORIGINAL

SUBSTITUTE SHEET

-13-

The circuit diagram for the amplifier mounted on board 32 is also detailed in Fig. 4. Pins P1, P2 and P3 represent separate connections of jack 37. Pin 1 is connected to a positive voltage source of typically 12 volts (the voltage may range from 8 to 15), P2 carries the output signal from the amplifier and P3 is connected to ground. Resistors R4 and R5, form a voltage divider so that relative to the DC reference for U1, namely the junction of R4 and R5, both positive and negative DC voltages are available. Points M1 and M2 are connected to the two leads of the transducer 31. Resistor R1 drains off static charges across the transducer and U1. R1 may be in the range from 220 k ohms (k = thousand) to 2.2 megohms. U1 is preferably a low noise field effect transistor (fet) input op amp such as the CA3140, Intersil 7611, or TL071 and is operated here as a non-inverting amplifier to provide high input impedance, essentially that of R1. The gain of U1 is thus the 1 plus ratio of resistances R3/ R1. A novel added capacitor C1, together with R2 and R3 provides a low pass filter to eliminate noise above the frequency, range of the fetal heart sound. For a 3 decibel (db) cutoff of about 160 Hz, R3 may be 1 megohm and C1 1000 micro-micro farads or picofarads (pf). The cutoff frequency is given by the formula cutoff frequency (Hz.) =  $1/(6.28 * C1 * R3)$ , where C1 is measured in microfarads and R3 in megohms. The cutoff frequency should be in the range 100 - 500 Hz., preferably in the range 120 - 250 Hz.

Typical component values (with ranges in parentheses) may be:

R1	2.2 megohms	(220 k ohms - 3 megohms)
R2	10 k ohms	(2.2 k - 22 k ohms)
R3	1 megohm	(100 k ohms - 2.2 megohms)

The ratio R3/R2 should be between 10 to 1 and

BAD ORIGINAL

SUBSTITUTE SHEET

400 to 1, and preferably between 50 to 1 and 100 to 1.

R4 6.8 k ohms (2.2 k - 10 k ohms)

R5 2.7 k ohms (2.2 k - 10 k ohms)

C1 1000 pf (300 pf - 5000 pf)

The value of C1 should be chosen to provide the desired cutoff frequency in conjunction with R3.

This novel design combines the usual active audio filter (6 db/ octave cutoff, here limited to a typical maximum of 20 db) and operational amplifier, using only one op amplifier.

The flexible attachment of microphones, absence of wires other than those connecting the microphones to the transmitter, and light weight and small size facilitate maternal movement and comfort. The use of multiple microphones insures that an adequate filtered, enhanced composite signal can be formed through later processing. Although normally the composite signal will be superior to any single signal, and no single signal need suffice, an adequate composite signal will be generated whenever one microphone detects a strong, relatively noise-free fetal heartbeat. In general, the aggregate signal-to-noise ratio of the fetal heart input signals as well as the signal-to-noise ratio of the final enhanced composite signal depend inversely upon the number of microphones. Provision is made to vary the number as needed depending on circumstances.



Referring to Fig. 5, <sup>-15-</sup> the signal from the microphone amplifier of Fig. 4 is further amplified, filtered, and demodulated. The signal enters via pin P2 and is coupled to the first stage of amplification U2 via capacitor C10. Resistor R10 provides a suitable load to the microphone amplifier U1, and minimizes AC hum by providing a relatively low impedance (typically 10 k ohms, the value might range from 2 k to 100 k ohms) path to ground for any induced voltages.

Resistor R11 maintains the junction of C10 with inverting input of U2 at DC ground. C10 may be a 1 microfarad tantalum capacitor, correctly polarized, and R11 may be 100 k ohms. This resistor-capacitor network also eliminates the need for offset null adjustment in the high-gain amplifier U1 and provides high pass filtering to the input signal. In this case the high pass filter has a 6 db per octave rolloff below a 3 db cutoff of 3 Hz. Other values of C10 and R11 may be used provided that R11 is sufficiently (typically 2 or more times) larger than R10. The operational amplifier U2, together with associated resistors R12, R13, and R14, and capacitor C11 is operated as a non-inverting amplifier combined with a low pass filter, as in the microphone amplifier U1. The values of C11 and R14 may be chosen so as to provide a 3 db cutoff frequency of 160 Hz and a 6 db per octave rolloff to a gain of 1. The cutoff frequency should be the same as that of the microphone amplifier of Fig. 4 in order to provide the smoothest pass-band and best filtering. Typical component values (with ranges in parentheses) associated with U2 are:

C11	.01 microfarad (.0047 - .047 microfarad)
R12	10 k ohms, variable (2.2 k - 250 k ohms, variable)
R13	270 ohms (100 - 5 k ohms)

-16-

R14 100 k ohms (10 k ohms - 1 megohm)

The above capacitor and resistors control the gain and filtering of this stage of amplification.

U2 T1071 or CA3140 or 741 type.

These values provide adjustable gain from 10 to about 400. Provision may be made for lower gain on the maternal microphone channel, typically a range from 1 to 40. Provision may also be made for slow automatic gain control using standard circuits. This automatic gain control can operate before or preferably after U2. U1 or U2 or both may be replaced by multiple, successive stages of amplification to provide smoother control of gain, both manually and automatically.

The resulting filtered, amplified signal from U2 is demodulated by the precision rectifier formed by U3 and U4 and associated components. This precision rectifier yields an output signal (junction of R17 and R20) equal to the absolute value of its input signal. The design of the precision rectifier is taken from Jung (1983) except that both positive and negative supply voltages are needed to prevent latchup. Typical component values and ranges here are:

R16, R17 100 k ohms, matched to .1 %

(10 k ohms - 1 megohm, as well  
matched as practical)

R18 27 k ohms (10 k - 100 k ohms)

R19 47 k ohms (10 k - 100 k ohms)

D1 1N914, 1N458A, or similar

U3, U4 each 1/2 358 or 1/2 TL072

This output is then sent through an inverting amplifier U5 operated as an active low passfilter with a cutoff frequency of 15 - 60, and 6 db rolloff to a gain of almost 0. The cutoff frequency should be less than half of

the sampling rate used in later data acquisition in order to prevent aliases. Several such filters may be used if desired to provide sharper cutoffs. The standard design here is based on Wyss (1984). Typical component values are:

C12 .047 microfarad (.0022 - 1 microfarad)

R20, R21 100 k ohms, matched to .1 %

(10 k ohms - 1 megohms, as well  
matched as practical)

U5 TL071 or 741 type

The output at P12 is sent to the analog data acquisition unit of the processing unit. R22 (2.2 k ohms - 100 k ohms, typically 10 k ohms) provides a low impedance path to ground for any induced voltages on the line from P22. This signal is a demodulated, filtered signal obtained from the fetal heart signal (or maternal heart signal) in the range 0 to about 5 volts.

Finally, U6 operates as a comparator to indicate levels. Light emitting diode (led) D2 lights whenever the signal exceeds the applied voltage at P11. Two comparators may be used; lighting a green led at adequate signals (above, say, .5 volt), and a red led on peaks (above, say, 4 volts). U6 may be a 741 or TL071 type, R22 2.2 k (1 - 10 k) ohms, and D2 any low power led.

The chip count may be reduced by using double or quad op amplifiers, taking the usual care with signal isolation. In addition the usual care should be taken to provide suitable paths to ground for any induced voltages on lines carrying signals, and to provide adequate bypassing.

Preferably the early amplification, through U2, is done before broadcasting, and the demodulation and filtering beginning at U3 is done after receiving the broadcast signal. Automatic gain control should be preferably provided in both

places. In this way the problem of accurate transmission and reception of a signal modulated by a very low frequency (2-4 Hz) signal is avoided.

The output from U2 may also be sent to headphones or, preferably with further amplification to a loudspeaker to aid in placing the microphones. If desired the output from U2 may be used to modulate a signal in the range 200 - 1000 Hz., preferably about 700 Hz., using for example, the XR2206 integrated circuit and standard components. This shifts the fetal heart sound to a more readily audible frequency range, which is then sent to headphones or a loudspeaker.

Referring to Fig. 6, the transmitter 12 is designed to locally broadcast the plural microphone and auxiliary signals, preferably multiplexed and modulated on a single carrier, to a nearby receiver 21. The microphone plugs 37 fit jacks 41 and the auxiliary input plugs fit jacks 42. Slowly varying auxiliary inputs may be encoded as audio-frequency signals via optional audio modulators (encoders) 43, or broadcast as received. All these signals from 41 and 42, whether passing through 43 or not, are amplified by amplifiers 44 to obtain suitable amplitudes for later modulation. These amplifiers may be the amplifiers of Fig. 5, including the filters therein. The demodulation is preferably not done at this stage, but is done after reception of the broadcast. The gain of the amplifiers may be set, and optionally, but preferably, controlled by suitable automatic gain controls for calibration purposes and to maintain adequate modulation and need not be readjusted during operation.

All of the separate signals are then sent to a single multiplexer 46 which contains a separate modulator strip 47 for each input signal. Each modulator strip contains a low-pass audio filter 48, passing signals unattenuated below a cutoff frequency chosen to prevent mutual interference,

preferably in the range 250-1000 Hz to balance the needs of adequate bandwidth for later processing and minimum total bandwidth for effective transmission. Each strip further contains a local oscillator 49, with the separate intermediate frequencies spaced at suitable intervals, a frequency modulator 50 to minimize noise, a buffer 51 and an output band-pass filter 52 centered about the intermediate frequency. The oscillators include suitable frequency standards such as phase-locked loops or crystals, and suitable buffers to maintain frequency stability. The filters 48 and 52, and spacing between intermediate frequencies are chosen so that each signal is subject to essentially no interference from other signals (unwanted signals should be at least 20, better at least 30, and preferably at least 40 decibels down in view of the accuracy of later digital processing). The required spacing between intermediate frequencies is typically 5-10 times the bandwidth of the filters 48. A wide range of intermediate frequencies is possible.

The separate modulated intermediate-frequency signals from 52 are combined at a intermediate-frequency combiner 56 into a single relatively broad-band multiplexed intermediate frequency signal. The combiner may be a summing amplifier or a hybrid device, together with suitable buffers and intermediate-frequency transformers.

In an alternate embodiment, the multiplexer 46 operates in the audio-frequency range using analogous audio-frequency components at 48, 49, 51, 52 and 56. In this the oscillators 49 operate at audio frequencies spaced by several times the cutoff frequencies of filters 48, using amplitude modulators at 50, and thus obtaining audio frequency signals at different frequencies from the filters 52. The output from the audio combiner 56 is then frequency modulated (not shown) on a suitable intermediate-frequency signal obtained from a suitable intermediate-frequency

oscillator.

The intermediate-frequency signal is then heterodyned to a suitable output frequency using a local oscillator 62, mixer 63, buffer/ amplifier 64, and filter 65. This may be done more than once if desired. The output from 65 then passes to a low power final amplifier 67 and filter 68, and then to the transmitting antenna 14. Output frequencies should be chosen, preferably in the ultra-high frequency range, so that local broadcasting can be obtained with very low power and noise and a very short flexible wire antenna 14.

The transmitter 12 is enclosed in a small plastic box, which may measure approximately 2 x 7.5 x 10 cm. attached to a light cloth belt (13 in Fig. 1) worn by the mother. The transmitter also contains conventional batteries, switches and controls, and test-points as required for positioning the microphones, maintenance, etc. For example, it may be desirable to allow for monitoring each amplified input signal for such purposes, or to aid in positioning the microphones 11. No adjustments are needed for normal operation on the patient.

The transmitted signal from Fig. 6 is picked up by a receiving antenna 16 connected to a receiver 21 in Fig. 7. The purpose of the receiver is to detect and demultiplex the signal broadcast from the mother's transmitter, obtaining the separate plural signals sent to the transmitter. The receiver contains the standard radio-frequency amplifier 81, mixer 82, oscillator 83, and intermediate frequency amplifier 84 to produce an intermediate frequency signal. (One or more oscillators, mixers, and intermediate frequencies may be used, in which case output from 84 represents the final such signal.)

The intermediate frequency signal from 84 is then

sent to a detector and demultiplexer 86 to recover the separate original input signals. Referring to Fig. 8 for details of the detector and demultiplexer 86, this is accomplished by first splitting the intermediate frequency signal into multiple separate signals, one corresponding to each input, using band-pass filters 88 and isolating amplifiers at either or both of 89 and 90, as desired. Each separate signal from 90 is detected at 92 with a suitable detector (FM in case frequency modulation was used). Any audio frequency signals used to encode slowly varying inputs at 43 in Fig. 6 are decoded at this point (not shown) to recover the original signals. These signals are amplified at amplifiers 93 and carried to the processor 22 by wires 94. The amplifiers 93 may be the amplifiers of Fig. 5 or similar units, including the filtering therein. These amplifiers also preferably include demodulators and low-pass filters for preconditioning of the signals. These may be the precision rectifiers and active filters of Fig. 5.

An alternate embodiment of the detector and demultiplexer 86 is required if the alternate multiplexer is used. In this embodiment, referring to Fig. 9, the signal from I-F amplifier 84 is detected by a suitable FM detector 92, amplified at amplifiers 93, and split into separate component signals using balanced mixers 97, audio-frequency oscillators 98 and filters 99. The signals from 99 are then amplified, if desired at amplifiers 100, and then carried to the processor 22 by leads 94.

Referring now to Fig. 2, the processor 22 produces a single enhanced fetal heartbeat signal from multiple fetal inputs and one maternal input, and also produces the power spectrum, or range of frequencies present in the fetal heartbeat and their amplitudes, as one measure of variability. It also accepts any other desired signals, such as tonometric, electromyographic or other data on contractions, maternal electrocardiographic data, etc.,

-22-

corresponding to auxiliary inputs 15.

These signals are sent to output devices 18 such as loudspeakers, oscilloscopes, strip chart recorders, indicator lights, etc. Further processing to find such data as instantaneous heart rates is also performed and the results displayed at 18. These output devices show both the fetal heartbeat sound signal, and its frequency spectrum for one measure of variability, as well as auxiliary inputs. It is also possible to receive signals from several transmitters at a central monitoring station.

#### Processing unit

Note that only major electronic components such as analog to digital (A/D) converters, microprocessors and RAM units are shown in Fig. 10. Other components such as latches, buffers, most interfaces and controllers, etc. depend upon the specific major components used, are not shown in Fig. 10, but the components required and their use will be clear to those skilled in the art. The processing unit is designed to execute the required processing and programs so that the enhanced fetal heart signal and frequency data lag only slightly behind real time.

The output signals from the receiver enter the processing unit by wires 94. The preferably demodulated fetal heart signals and a maternal heart signal used in the filtering process are then processed.

In the event the particular demodulation technique used causes two beats per cycle, as with some demodulators, one must recognize that the frequencies in the power spectrum described below are twice the actual heart rates, and divide by two, accordingly.



Each fetal heart signal as well as the maternal heart signal is separately digitized and sampled at a separate analog data acquisition unit 301. (For simplicity, details of only two of these modules are shown.) Each analog data acquisition unit contains a linear amplifier 123 with signal conditioning circuitry, a sample and hold unit 125, an analog to digital (A/D) converter 127 and an interface 129. (The units 301 may be combined in an input/output package, together with digital to analog (D/A) converters 181 if desired. Alternatively, a single unit 301 may be used to sample several signals provided the signals are first appropriately time-multiplexed, as, for example in the Intel iSBX 311.) In this process, each signal is first sent to a separate linear amplifier 123. The amplifiers may so constructed as to filter out signals at frequencies above the sampling rate, using resistor-capacitor, active, inductor-capacitor, or other similar filters. Active filters may be similar to those of Fig. 5. This is a standard preconditioning technique familiar to those skilled in the art of digital sampling. These signals are then sent to separate sample and hold units 125 under the control of microprocessor 145, which controls the sampling process, via interface 129. The A/D converters 127 digitize the signals temporarily stored in the sample and hold units 125, under the control of microprocessor 145 through interface 129.

The output of each A/D convertor is then sent to a separate dual-ported random access memory (RAM) unit 131 through interface 129 under the control of microprocessor 145. (If desired, data may instead pass through the microprocessor or direct memory access techniques and circuitry may be used. Similar observations apply to all dual-ported RAM units.) The microprocessor may be an 8 bit or preferably a 16 bit unit such as the 68000, 80186, or 16032. The outputs from the A/D convertors should be 8-16 and preferably 12-16 bits wide. The RAM is most easily

configured into words of corresponding numbers of bits. All required timing is controlled by one or more clocks 144. Signals from the auxiliary inputs, including a second copy of the maternal heart signal if desired, are sampled and stored in similar dual-ported RAM units to slightly delay these signals for synchronization with the processed fetal heart signal. The choices of sampling rate, number of samples per cycle, and duration of sampling cycle all contribute to effective operation.

First, detection of the fetal heart signal without aliases requires a sampling rate at the Nyquist frequency, at least twice the expected rate of beats, or four times the maximum heart rate, accounting for two beats per cycle. This should be significantly increased, preferably to at least the Nyquist frequency of the fourth and preferably the fifth harmonic of the fetal heart rate to provide accurate measurements and waveforms. Thus a sampling rate of at least 1800 samples per minute, and preferably at least 3600 samples per minute should be used. The latter rate implies that even at a fetal heartrate of 180, the fifth harmonic can be detected without aliasing. Even higher rates such as 7200 samples per second, are better.

In addition, if good information about audio components up to 100 Hz is required, the sampling rate must be above twice 100 Hz, i.e., at least 12000 times per minute.

Accurate measurement of variability from the enhanced fetal heart signal is also favored by a high sampling rate. For example, an accuracy of  $\pm 1$  beat per second requires a sampling rate of at least 3600, and preferably at least 7200 samples per minute.

Secondly, "60-cycle" electrical noise (in the United States) is most effectively eliminated in later processing when the sampling rate bears a simple harmonic relation to 60

Hz. Thus rates of 1800, 3600, 7200, etc., samples per minute are preferred. (In European or other versions, the sampling rate should have a similar relation with the electrical frequency in use, e.g. 1500, 3000, or 6000 samples per minute for 50 Hz current.)

Processing requirements will generally limit the sampling rate to a typical rate of 3600 or 7200 samples per minute, as described below. Sampling rates of 7200 samples per minute can be obtained and processed with chips such as the 68000 or 80186 or at 14400 with the 68020, coprocessors or dedicated chips.

In the FFT, the number of samples per sampling cycle must be a power of two.

The sampling cycle is limited by two considerations. The first consideration, maximizing accuracy, leads to longer sampling cycles. Simulation suggests that cycles of at least 4 seconds are required, cycles of at least 8 seconds are better and yield an accuracy of about  $\pm 1.5$  beats per minute, and cycles of 16 or more seconds are more accurate and thus preferred.

The second consideration, minimizing the time lag of two sampling cycles suggests shorter sampling cycles. At the upper end, the delay in a 82 second cycle would be 2.75 minutes, or about the interval between onsets of contractions. Shortening of the sampling cycle also allows more accurate computation of short term variability, in addition to shortening the delay. In addition, since the enhanced fetal heart signal is constructed adaptively with each sampling cycle, shorter cycles also favor more effective noise elimination in this manner.

Both considerations suggest that periods of 16-35 seconds are preferred. This may be controlled by a switch

(not shown) connected to microprocessor 145, or if desired, automatically. In this case, periods as short as 4-9 seconds (and preferably 8-17 seconds) might be selected to measure acceleration or short term variability, and cycles as long as 75 seconds might preferably be used to gather the most accurate data during a contraction. In addition, cycles might be triggered by the start of contractions if desired.

-27-

These considerations yield a range of practical and preferred sampling rates and patterns:

Samples per minute	Samples per sampling cycle	Length of sampling cycle in seconds	Relative quality
1800	128	4.27	possible
	256	8.53	possible
	512	17.07	possible
	1024	34.34	possible
	2048	68.68	possible
3600	256	4.27	possible
	512	8.53	possible
	1024	17.07	good
	2048	34.34	good
	4096	68.68	possible
7200	512	4.27	possible
	1024	8.53	good
	2048	17.07	better
	4096	34.14	better
	8192	68.28	good
14400	2048	8.53	good
	4096	17.07	better
	8192	34.14	better
	16384	68.28	good

All times are rounded to .01 second. Exact times in seconds are 60 times the quotient (samples per sampling cycle) / (sampling rate). Analogous European rates are found by multiplying US rates by 5/6, corresponding European sampling cycles are 1.2 times as long as US cycles.

Microprocessor 145 controls the sampling rate and sampling cycle by controlling the analog data acquisition units 301, the dual-ported RAM units 131, and dual-ported RAM units (not shown) used to delay auxiliary inputs, as described below.

Although Figs. 8 and 9 show three microprocessors 145, 155, and 170, to illustrate three main functions, namely input, processing, and output, more or fewer microprocessors may be used.

To effect the conversion of the digitized fetal heartbeat signals to the frequency domain, in which signals are represented in terms of the frequencies present and the amplitude at each frequency, an algorithm called the FFT is used. The FFT is illustrated by lines 1000-1900 of the following program, written in Basic, and is an improved novel version of the standard algorithm as found in Horowitz and Sahni (1978), see also Aho, Hopcroft, and Ullman (1974). There are many variants on the FFT algorithm. (The complete program is given here for later reference.)

```
1000  REM  MODULE TO COMPUTE FFT
1010  REM  LINES 1000-1900 ILLUSTRATE BLOCK 312 OF FIG. 11.
1020  REM      LINES 1000-1900 ARE AN IMPROVED VERSION
      (ESPECIALLY IN BIT REVERSALS OF THE HOROWITZ AND SAHNI (1978)
      ALGORITHM
1050  REM  FLOATING POINT ARITHMETIC IS USED IN THIS PROGRAM;
      UNLESS FAST FLOATING POINT OPERATIONS ARE AVAILABLE IT IS
      REPLACED BY INTEGER ARITHMETIC USING STANDARD DESCRIPTIONS OF
      FLOATING POINT AND FIXED POINT ARITHMETIC.
1060  REM  PRELIMINARY STEPS
1070  INPUT M: REM  NUMBER OF DATA POINTS IS  $N = 2^M$ .
1080  REM      WHENEVER SPECIFIC NUMBERS ARE SHOWN FOR
      SUBSCRIPTS,  $M=11$  IS ASSUMED.
1090   $N = 2 ^ M$ 
1100   $NQ = 2 * N$ 
1110  DIM  AI(5,N),A(5,N,2),INDEX(N): REM  EACH VECTOR
      (A(MC,I,1),A(MC,I,2)) IS INTERPRETED AS A COMPLEX NUMBER
1120  REM  MC IS MICROPHONE NUMBER
1130  DIM  R(2),S(2),T(2): REM  THESE VECTORS REPRESENT
      COMPLEX NUMBERS
1140  REM  INPUTS ARE OBTAINED FROM D/A CONVERTER MODULES 301
1150  FOR I = 1 TO N
1160  FOR MC = 1 TO 5: REM  MC 1 TO 4 REPRESENT FETAL HEART
      INPUTS AND MC 5 REPRESENTS MATERNAL HEART INPUT.
1170  INPUT AI(MC,I): REM  ITH DATA POINT FROM UNIT MC
1180  NEXT MC
1190  NEXT I
1200  REM  DESCRIPTION OF BIT REVERSAL ACCOMPLISHED IN
      HARDWARE
1210  REM  THE BIT REVERSAL IS ACCOMPLISHED BY REVERSING THE
      ORDER OF THE SEPARATE ADDRESS LINES IN BUS 272 OF DUAL-PORTED
      RAM 131.
1220  REM  IN THE ACTUAL IMPLIMENTATION, EACH ENTRY A(I) IS
```

STORED IN MEMORY LOCATION LOC(I)=I-1.

1230 REM THE BIT REVERSAL REPLACES LOC(I) BY REV(LOC(I)),  
THE BINARY INTEGER OBTAINED BY WRITING LOC(I) AS AN M-PLACE  
BINARY INTEGER, INCLUDING LEADING ZEROS, AND REVERSING

1240 REM (CONTINUED) OF THESE DIGITS TO OBTAIN REV(LOC(I)).  
INDEX(I) = REV(LOC(I))+1. BIT REVERSAL IS ITS OWN INVERSE  
OPERATION.

1250 INDEX(1) = 1

1260 LMAX = 1

1270 II = N / 2

1280 FOR K = 1 TO M

1290 FOR L = 1 TO LMAX

1300 INDEX(L + LMAX) = INDEX(L) + II

1310 NEXT L

1320 II = II / 2

1330 LMAX = 2 \* LMAX

1340 NEXT K

1350 FOR I = 1 TO N

1360 FOR MC = 1 TO 5

1370 A(MC, INDEX(I), 1) = AI(MC, I)

1380 NEXT MC

1390 NEXT I

1400 REM END OF BIT REVERSAL

1410 REM FFT AFTER BIT REVERSAL

1420 REM AT THE END, WHERE COMPLEX NUMBERS ARE REPRESENTED BY  
VECTORS, THE VECTOR (A(MC, I, 1), A(MC, I, 2)) IS THE FOURIER  
COEFFICIENT ASSOCIATED WITH  $(\cos(2\pi I/N), \sin(2\pi I/N)^{(I-1)})$ .

1430 HN = N / 2

1440 PI = 3.14159265: P = 2: HP = 1

1450 FOR L = 1 TO M

1460 ANGLE = 0

1470 THETA = PI / HP

1480 FOR J = 1 TO HP

1490 FOR I = J TO N STEP P

1500 IO = I + HP

1510 C = COS (ANGLE): S = SIN (ANGLE): REM ANGLE =  
(J-1)\*PI/HP. COS(ANGLE) AND SIN(ANGLE) ARE THE  
(J-1)\*N/(2\*HP) TH ENTRIES IN RESPECTIVE TABLES IN TRIG FCN



ROM OF COS AND SIN FOR ANGLES FROM 0 TO  $(N/2 - 1) * 2 * \pi / N$  IN STEPS OF  $2 * \pi / N$ .

1520 REM THE NEXT FOUR STATEMENTS PERFORM FAST COMPLEX MULTIPLICATION  $(A(MC, I0, 1), A(MC, I0, 2)) * (C, S)$ . IF A FAST MULTIPLY IS AVAILABLE, THEN THEY SHOULD BE REPLACED BY THE ORDINARY IMPLIMENTATION OF COMPLEX MULTIPLICATION.

1530 FOR MC = 1 TO 5

1540 V1 = A(MC, I0, 1) \* C

1550 V2 = A(MC, I0, 2) \* S

1560 T(1) = V1 - V2

1570 T(2) = (A(MC, I0, 1) + A(MC, I0, 2)) \* (C + S) - V1 - V2

1580 A(MC, I0, 1) = A(MC, I, 1) - T(1): A(MC, I0, 2) = A(MC, I, 2) - T(2)

1590 A(MC, I, 1) = A(MC, I, 1) + T(1): A(MC, I, 2) = A(MC, I, 2) + T(2)

1600 NEXT MC

1610 NEXT I

1620 ANGLE = ANGLE + THETA

1630 NEXT J

1640 HP = P:P = 2 \* P

1650 NEXT L

1660 REM CALCULATIONS COMPLETE; PRINT RESULTS FOR ILLUSTRATION.

1670 PRINT "FFT RESULTS"

1680 REM OUTPUT FROM FFT IS COMMONLY CALLED THE SPECTRUM.

MC 1 TO MC 4 REPRESENT SPECTRA FROM FETAL HEART SIGNALS AND

MC 5 REPRESENTS THE SPECTRUM FROM THE MATERNAL HEART SIGNAL.

1690 FOR I = 1 TO N

1700 FOR MC = 1 TO 5

1710 PRINT MC, I, A(MC, I, 1), A(MC, I, 2)

1720 NEXT MC

1730 NEXT I

1900 REM END OF MODULE TO COMPUTE FFT

2000 REM MODULE TO FILTER OUTPUT FROM FFT

2010 REM ILLUSTRATES BLOCK 321 OF FIG. 11.

2020 REM LINES 2500-2900 ILLUSTRATE OPTIONAL FILTERING TO REMOVE NOISE IN FETAL HEART SPECTRA DUE TO MATERNAL HEARTBEAT.

2030 REM LINEAR FILTER ILLUSTRATED

2040 REM A WIDE CHOISE OF LINEAR AND NON-LINEAR FILTERS IS POSSIBLE.

2050 REM FFC STORES FETAL FILTER COEFFICIENTS FOR SIMPLE LINEAR FILTER

2060 REM MFC STORES MATERNAL FILTER COEFFICIENTS FOR SIMPLE LINEAR FILTER

2070 REM THESE VARIABLES ARE USED FOR CLARITY.

2080 REM IN THE ACTUAL IMPLIMENTATION,MULTIPLICATION BY 0 IS REPLACED BY ASSIGNMENT OF 0 AND MULTIPLICATION BY 1 IS NOT PERFORMED.

2085 REM IN THE ACTUAL IMPLIMENTATION, FFC AND MCF COEFFICIENTS ARE SET ONCE BEFORE PROCESSING. THEY ARE DESCRIBED HERE FOR CLARITY.

2090 REM TO ELIMINATE 60 HZ NOISE, MAKE THE CORRESPONDING FILTER COEFFICIENTS = 0. AT A TYPICAL SAMPLING RATE OF 7200 SAMPLES PER MINUTE (120/ SEC.) AND A SAMPLING CYCLE OF 2048 SAMPLES (17.07 SEC.), SET FFC(1025) = 0 AND MFC(1025) = 0.

2100 REM AT A TYPICAL SAMPLING RATE OF 120 SAMPLES PER SECOND, AND A SAMPLING SYCLE OF 2048 SAMPLES OR 17.07 SECONDS, THIS REQUIRES THAT FFC(1025) = 0 AND MFC(1025)=0.

2110 REM OTHER FFC COEFFICIENTS SHOWN EMPHASIZE RATES OF 60 - 240 BEATS PER MINUTE. OTHER RANGES, E.G., 80 - 160 BEATS PER MINUTE ARE EASILY EMPHASIZED.

2120 REM OTHER MFC COEFFICIENTS SHOWN EMPHASIZE RATES OF 50 - 150 BEATS PER MINUTE. OTHER RANGES, E.G., 55 - 135 BEATS PER MINUTE ARE EASILY EMPHASIZED.

2125 DIM FFC(N)

2130 FOR I = 1 TO 9:FFC(I) = 0:FFC(2049 - I) = 0: NEXT

2140 FOR I = 10 TO 24:FFC(I) = (I - 9) / 16:FFC(2049 - I) = (I - 9) / 16: NEXT

2150 FOR I = 25 TO 341:FFC(I) = 1:FFC(2049 - I) = 1: NEXT

2160 FOR I = 342 TO 468:FFC(I) = (469 - I) / 128:FFC(2049 - I) = (469 - I) / 128: NEXT

2170 FOR I = 469 TO 1580:FFC(I) = 0: NEXT

2180 DIM MFC(N)

2190 FOR I = 1 TO 7:MFC(I) = 0:MFC(2049 - I) = 0: NEXT

2200 FOR I = 8 TO 14:MFC(I) = (I - 7) / 8:MFC(2049 - I) = (I

```
- 7) / 8: NEXT
2210 FOR I = 15 TO 214:MFC(I) = 1:MFC(2049 - I) = 1: NEXT
2220 FOR I = 215 TO 341:MFC(I) = (342 - I) / 128:MFC(2049 -
I) = (342 - I) / 128: NEXT
2230 FOR I = 342 TO 1707:MFC(I) = 0: NEXT
2240 FOR I = 1 TO N
2250 FOR MC = 1 TO 4
2260 A(MC,I,1) = A(MC,I,1) * FFC(I):A(MC,I,2) = A(MC,I,2) *
FFC(I)
2270 NEXT J
2280 A(5,I,1) = A(5,I,1) * MFC(I):A(5,I,2) = A(5,I,2) *
MFC(I)
2290 NEXT I
2500 REM FILTER FETAL SPECTRA A(1,-,-) THROUGH A(4,-,-) BY
FORMING THEIR PROJECTIONS ORTHOGONAL TO THE MATERNAL SPECTRUM
A(5,-,-). IF DESIRED, THE AMOUNT OF FILTERING MAY BE REDUCED
BY MULTIPLYING DP(1) AND DP(2) BY A FACTOR LESS THAN 1.
2510 DIM DP(2): REM HOLDS DOT PRODUCTS OF COMPLEX VECTORS
2520 REM CALCULATE SQUARE OF NORM OF A(5,-,-), OMITTING
CONSTANT TERM (A(5,1,-))
2530 NA = 0
2540 FOR I = 2 TO N
2550 NA = NA + A(5,I,1) * A(5,I,1) + A(5,I,2) * A(5,I,2)
2560 NEXT I
2570 FOR MC = 1 TO 4
2580 REM CALCULATE DOT PRODUCT OF A(MC,-,-) AND A(5,-,-),
OMITTING CONSTANT (FIRST) ENTRY
2590 DP(1) = 0:DP(2) = 0
2600 FOR I = 2 TO N
2610 REM FAST COMPLEX MULTIPLICATION
2620 V1 = A(MC,I,1) * A(5,I,1)
2630 V2 = - A(MC,I,2) * A(5,I,2)
2640 DP(1) = DP(1) + V1 - V2
2650 DP(2) = DP(2) + (A(MC,I,1) + A(MC,I,2)) * (A(5,I,1) -
A(5,I,2)) - V1 - V2
2660 NEXT I
2670 DP(1) = DP(1) / NA
2680 DP(2) = DP(2) / NA
```

```
2690 REM FORM ORTHOGONAL COMPLEMENT
2700 FOR I = 1 TO N
2710 REM FAST COMPLEX MULTIPLICATION OF DP TIMES A(5,I,-)
STORED IN T(1),T(2)
2720 V1 = DP(1) * A(5,I,1)
2730 V2 = DP(2) * A(5,I,2)
2740 T(1) = V1 - V2
2750 T(2) = (DP(1) + DP(2)) * (A(5,I,1) + A(5,I,2)) - V1 - V2
2760 REM REMOVE MATERNAL (A(5,-,-)) COMPONENT
2770 A(MC,I,1) = A(MC,I,1) - T(1)
2780 A(MC,I,2) = A(MC,I,2) - T(2)
2790 NEXT I
2800 NEXT MC
2900 REM END OF MODULE TO FILTER OUTPUT FROM FFT.
3000 REM MODULE TO EVALUATE QUALITY OF EACH FILTERED
FETAL HEART SPECTRUM.
3010 REM ILLUSTRATES BLOCK 316 OF FIG. 11.
3020 REM HN=N/2
3030 DIM PS(4,HN + 1): REM POWER SPECTRUM. IN PRACTICE
SMALLER VECTORS ARE USED BECAUSE THE FILTERING PROCESS
ELIMINATED MANY COMPONENTS OF THE SPECTRUM AND THUS OF THE
POWER SPECTRUM.
3040 REM THE ELIMINATED COMPONENTS NEED NOT BE COMPUTED OR
STORED.
3050 REM PS(MC,I) HOLDS THE (I-1)ST COMPONENT, NAMELY THAT
OF I-1 CYCLES PER SAMPLING PERIOD, OF THE POWER SPECTRUM OF
THE FILTERED SIGNAL OF MICROPHONE I
3060 FOR MC = 1 TO 4
3070 FOR I = 2 TO HN
3080 KT = N + 2 - I
3090 PS(MC,I) = A(MC,I,1) * A(MC,I,1) + A(MC,I,2) * A(MC,I,2)
+ A(MC,KT,1) * A(MC,KT,1) + A(MC,KT,2) * A(MC,KT,2)
3100 NEXT I
3110 PS(MC,1) = A(MC,1,1) * A(MC,1,1) + A(MC,2,2) * A(MC,2,2)
3120 PS(MC,HN + 1) = A(MC,HN + 1,1) * A(MC,HN + 1,1) +
A(MC,HN + 1,2) * A(MC,HN + 1,2)
3130 NEXT MC
3140 REM EACH FETAL HEART SIGNAL ENERGY IS ESTIMATED BY
```

ADDING COMPONENTS OF ITS FILTERED POWER SPECTRUM REPRESENTING A RANGE OF 1 TO 4 HZ (60 - 240 BEATS PER MINUTE). OTHER RANGES MAY BE USED.

3150 REM SNR IS THE VECTOR OF SIGNAL TO SIGNAL PLUS NOISE (TOTAL) ENERGIES FOR FILTERED FETAL HEART SPECTRA.

3160 DIM AC(N,2),SNR(4): REM AC STORES COMBINED SPECTRUM

3170 FOR MC = 1 TO 4

3180 SI = 0:EN = 0: REM RESPECTIVE ESTIMATED AMPLITUDE OF SIGNAL AND OF TOTAL SPECTRUM, OMITTING CONSTANT (FIRST) ENTRY IN EACH CASE

3190 FOR I = 2 TO 14: REM HN=N/2, 120 SAMPLES PER SECOND AND 2048 SAMPLES PER CYCLE ASSUMED HERE.

3200 KT = N + 2 - I

3210 EN = EN + PS(MC,I) + PS(MC,KT)

3220 NEXT I

3230 FOR I = 15 TO 45

3235 KT = N + 2 - I

3240 SI = SI + PS(MC,I) + PS(MC,KT)

3250 NEXT I

3260 EN = EN + SI

3270 FOR I = 46 TO HN: REM CAN STOP AS SOON AS PS(MC,I)=0.

3275 KT = N + 2 - I

3280 EN = EN + PS(MC,I) + PS(MC,KT)

3290 NEXT I

3300 IF EN = 0 THEN SNR(I) = 0: GOTO 3320

3310 SNR(I) = SI / EN

3320 REM CONTINUE

3330 NEXT MC

3900 REM END OF MODULE TO EVALUATE QUALITY OF EACH FILTERED FETAL HEART SPECTRUM.

4000 REM MODULE TO CALCULATE COMBINED, FILTERED SPECTRUM OF FETAL HEART SIGNALS.

4010 REM ILLUSTRATES BLOCK 323 OF FIG. 11.

4020 FOR I = 1 TO N

4030 AC(I,1) = SNR(1) \* A(1,I,1) + SNR(2) \* A(2,I,1) + SNR(3) \* A(3,I,1) + SNR(4) \* A(4,I,1)

4040 AC(I,2) = SNR(1) \* A(1,I,2) + SNR(2) \* A(2,I,2) + SNR(3) \* A(3,I,2) + SNR(4) \* A(4,I,2)

```
4050 NEXT I
4900 REM END OF MODULE TO CALCULATE COMBINED, FILTERED
SPECTRUM OF FETAL HEART SIGNALS.
5000 REM MODULE TO COMPUTE DISTRIBUTION OF VALUES OF FETAL
HEART BEAT.
5010 REM ILLUSTRATES BLOCK 318 OF FIG. 11.
5020 REM THAT PORTION OF THE POWER SPECTRUM REPRESENTING
HEART RATES OF APPROXIMATELY 60 - 240 BEATS PER MINUTE IS
FIRST COMPUTED AND THEN SMOOTHED.
5030 DIM AX(70),OP(69): REM HOLD PART OF POWER SPECTRUM
AND PART OF SMOOTHED POWER SPECTRUM, RESPECTIVELY.
5040 REM COMPUTE APPROPRIATE PART OF POWER SPECTRUM OF
ENHANCED, COMPOSITE FETAL HEART SPECTRUM.
5050 FOR I = 16 TO 70: REM BEATS PER MINUTE = 60 * (I-1) /
17.07
5060 KT = N + 2 - I
5070 AX(I) = AC(I,1) * AC(I,1) + AC(I,2) * AC(I,2) + AC(KT,1)
* AC(KT,1) * AC(KT,2) * AC(KT,2)
5080 NEXT I
5090 REM SMOOTH POWER SPECTRUM, OBTAINING SMOOTHED
SPECTRUM OP(-). SIMPLE LINEAR SMOOTHING ILLUSTRATED.
5100 FOR I = 17 TO 69
5110 OP(I) = (AX(I - 1) + 4 * AX(I) + AX(I + 1)) / 6
5120 NEXT I
5130 REM SEND RESULTS TO INTERFACE 210 AND THEN TO OUTPUT
DEVICES 211. PRINT HERE FOR ILLUSTRATION.
5140 FOR I = 17 TO 69
5150 PRINT 60 * (I - 1) / 17.07,OP(I)
5160 NEXT I
5900 REM END OF MODULE TO COMPUTE DISTRIBUTION OF VALUES OF
FETAL HEART BEAT.
6000 REM MODULE TO PERFORM INVERSE FFT
6020 REM ILLUSTRATES BLOCK 326 IN FIG. 11.
6050 REM FLOATING POINT ARITHMETIC IS USED IN THIS PROGRAM;
UNLESS FAST FLOATING POINT OPERATIONS ARE AVAILABLE IT IS
REPLACED BY INTEGER ARITHMETIC USING STANDARD DESCRIPTIONS OF
FLOATING POINT AND FIXED POINT ARITHMETIC.
6060 REM INVERSE FFT BEFORE BIT REVERSAL
```

-37-

```

6070 DIM B(N,2): REM B HOLDS OUTPUT
6080 P = N:HP = P / 2
6090 FOR L = M TO 1 STEP - 1
6100 ANGLE = 0
6110 THETA = PI / HP
6120 FOR J = 1 TO HP
6130 FOR I = J TO N STEP P
6140 IO = I + HP
6150 C = COS (ANGLE):S = SIN (ANGLE): REM ANGLE =
(J-1)*PI/HP. COS(ANGLE) AND SIN(ANGLE) ARE THE
(J-1)*N/(2*HP) TH ENTRIES IN RESPECTIVE TABLES IN TRIG FCN
ROM OF COS AND SIN FOR ANGLES FROM 0 TO (N/2 -1)*2*PI/N IN
STEPS OF 2*PI/N.
6160 T(1) = AC(I,1) - AC(IO,1):T(2) = AC(I,2) - AC(IO,2)
6170 AC(I,1) = (AC(I,1) + AC(IO,1)) / 2:AC(I,2) = (AC(I,2) +
AC(IO,2)) / 2
6180 REM FAST COMPLEX MULTIPLICATION, COMPUTING
(T(1),T(2))*(C,-S)/2 AND STORING RESULT IN
(AC(IO,1),AC(IO,2)). IF A FAST MULTIPLY IS AVAILABLE, THE
USUAL IMPLIMENTATION OF COMPLEX MULTIPLICATION SHOULD BE USED
INSTEAD.
6190 V1 = T(1) * C
6200 V2 = - T(2) * S
6210 AC(IO,1) = (V1 - V2) / 2
6220 AC(IO,2) = ((T(1) + T(2)) * (C - S) - V1 - V2) / 2
6230 NEXT I
6240 ANGLE = ANGLE + THETA
6250 NEXT J
6260 P = HP:HP = HP / 2
6270 NEXT L
6280 REM DESCRIPTION OF BIT REVERSAL ACCOMPLISHED IN
HARDWARE
6290 REM THE BIT REVERSAL IS ACCOMPLISHED BY REVERSING THE
ORDER OF THE SEPARATE ADDRESS LINES IN BUS 272 OF DUAL-PORTED
RAM 168.
6300 REM DETAILS ARE SIMILAR TO LINES 1200 THROUGH 1400,
ABOVE.
6310 FOR I = 1 TO N

```

```
6320 B(INDEX(I),1) = AC(I,1):B(INDEX(I),2) = AC(I,2)
6330 NEXT I
6340 REM CALCULATIONS COMPLETE
6350 REM B REPRESENTS OUTPUT FROM DUAL-PORTED RAM 168.
6360 REM PRINT FOR ILLUSTRATION
6370 PRINT "INVERSE FFT RESULTS"
6380 FOR I = 1 TO N
6390 PRINT I,B(I,1),B(I,2)
6400 NEXT
6900 END : REM END OF MODULE TO COMPUTE INVERSE FFT AND OF
PROGRAM
```



Except for the bit reversals shown in lines 1200-1400 if desired, the actual algorithm used may be implemented either as a machine language program stored in read only memory (ROM), or directly on a special purpose FFT chip 161.

Here is a second illustrative program, written in PL/M to run on the Intel 8086 based computers under RMX. Much of this program is similar to the Basic program above, as shown in the following table (Block numbers refer to Fig. 11, FFT denotes Fast Fourier Transform):

Module	Lines of PL/M program	Lines of Basic program
Declarations of external procedures	1 - 52	Not required
Interrupt procedure to acquire data	53 - 83	Not illustrated
Declarations and initial- ization	83 - 182	Usually not required. Otherwise, included where needed.
Detrend input data	183 - 203	Not included
FFT (Block 312)	204 - 230	1000-1900
Filter out- put from FFT (Block 321)	231 - 279	2000-2900

-40-

Evaluate quality of signals (Block 316)	280 - 303	3000-3900
Calculate combined, filtered spectrum (Block 323)	304 - 311	4000-4900
Compute distribution of values of fetal heart beat (Block 318)	312 - 328	5000-5900
Inverse FFT (Block 326)	329 - 370	6000-6900
Retrend data	Included in 329 - 370	Not illustrated
Print or save data, ask whether to run again	371 - 408	Not illustrated

-41-

Copyright 1985  
Harold M. Hastings  
Pages 41-64

```
$compact
1      efhmdata: do;

/* external procedures */

2      1      endout: procedure external;
3      2      end endout;

4      1      putint: procedure (outint) external;
5      2      declare outint integer;
6      2      end putint;

7      1      putskip: procedure external;
8      2      end putskip;

9      1      putstring: procedure (ptr) external;
10     2      declare ptr pointer;
11     2      end putstring;

12     1      getint: procedure integer external;
13     2      end getint;

14     1      writefile: procedure(dataptr, datalength) external;
15     2      declare dataptr pointer;
16     2      declare datalength word;
17     2      end writefile;
```

```

/* internal procedures */

18 1  expfcn: procedure(x) real;
19 2  declare (x,y,z) real;
20 2  y = 1.0+x*(1.0+(x/2.0)*(1.0+(x/3.0)
      *(1.0+(x/4.0)*(1.0+(x/5.0)*z)))));
21 2  z = 1.0+(x/6.0)*(1.0+(x/7.0)*(1.0+(x/8.0)
      *(1.0+(x/9.0)*(1.0+(x/10.0)*z)))));
22 2  return y;
23 2  end expfcn;

24 1  taper: procedure (i,j,k) real;
      /* uses counterexample to real analytic
      continuation to produce a smooth function
      which is 1 for k <= i and 0 for k >= j */
25 2  declare (i,j,k) integer;
26 2  declare (x,y,z,ii,jj,kk) real;
27 2  if k <= i then return 1.0;
29 2  else if k >= j then return 0.0;
31 2  else do;
32 3  ii = float(i);
33 3  jj = float(j);
34 3  kk = float(k);
35 3  z = (jj - ii)/2.5;
36 3  x = expfcn(-z*z/((kk-ii)*(kk-ii)));
37 3  y = expfcn(-z*z/((kk-jj)*(kk-jj)));
38 3  z = y/(x+y);
39 3  return z;
40 3  end;
41 2  end taper;

/*****
/* declare interrupt procedures */
*****/

42 1  DECLARE TOKEN LITERALLY 'SELECTOR';
43 1  rq$exit$interrupt: PROCEDURE (level,except$ptr)
      EXTERNAL;

```

```
44  2      DECLARE      level      WORD,
                                except$ptr  POINTER;
45  2      END rq$exit$interrupt;

46  1      rq$set$interrupt:  PROCEDURE( level,
                                interrupt$task$flag, interrupt$handler,
                                interrupt$handler$ds, except$ptr) EXTERNAL;

47  2      DECLARE level      WORD,
                                interrupt$task$flag  BYTE,
                                interrupt$handler    POINTER,
                                interrupt$handler$ds  TOKEN,
                                except$ptr            POINTER;

48  2      END rq$set$interrupt;

49  1      rq$reset$interrupt:  PROCEDURE( level,
                                except$ptr ) EXTERNAL;

50  2      DECLARE level      WORD,
                                except$ptr  POINTER;

51  2      END rq$reset$interrupt;
```

-44-

```

52  1  DECLARE interrupt$level$3      LITERALLY '3EH',
        interrupt$task$flag . BYTE,
        interrupt$handler      POINTER,
        data$segment          SELECTOR,
        status                 WORD;

        /*****
        /* interrupt procedure to get data */
        *****/

53  1  HANDLER: PROCEDURE INTERRUPT 128;

54  2  DECLARE HIGH LITERALLY '080H',
        /*port for high byte*/
        LOW LITERALLY '082H'
        /*port for low byte*/,
        ADR LITERALLY '080H';
        /*channel select address*/

55  2  DECLARE ADR2 LITERALLY '0D2H',
        /* address of timer 0 */
        ADR6 LITERALLY '0D6H',
        /* address for timer mode control word */
        MCR LITERALLY '01100000B',
        /* mode control word */
        /* 01 selects counter 1
           10 selects read/write msb
           000 selects mode 0 (interrupt)
           0 selects hex timer */
        COUNTS LITERALLY '00AH'; /* count for timer */

56  2  DECLARE (X,Y,mc,FLAG) BYTE; /* dummy */

57  2  SAMPLEDONE = 0;

```

SUBSTITUTE SHEET

```
58 2  OUTPUT(ADR6) = MCR;
      /*load mode control word into timer*/
59 2  X = INPUT(HIGH);
      /*read high byte to ensure EOC bit reset*/
60 2  OUTPUT(ADR2) = COUNTS;
      /* initial count for timer 0 */

61 2  do mc = 0 to maxmc;
62 3  OUTPUT(ADR) =mc; /* select channel mc */
63 3  FLAG = 1;
64 3  CHECKEOC: DO WHILE FLAG = 1;
      /* check for EOC, end of conversion */
65 4  FLAG = INPUT(LOW) AND 01H;
      /* flag = 0 bit of low byte = EOC bit */;
67 4  END CHECKEOC; /* flag goes low? on EOC */

68 3  X = INPUT(LOW);
      /* read low byte of channel MC */
69 3  X = X AND 0FOH;
      /*and mask off non-data (4 low) bits*/
70 3  Y = INPUT(HIGH);
      /* read high byte of channel MC */
71 3  INDATA(PT+woffset).mic(mc)
      = INT(X/010H+010H*Y);
      /* combine data bits of low byte with high
      byte to obtain 12 bit data */
      /* end of one conversion */
72 3  end;

73 2  IF PT = N THEN DO;
75 3  SAMPLEDONE = 1;
76 3  PT = 0;
77 3  IF PROGDONE = 0 THEN EEKFLAG = 1;
79 3  END;
80 2  PT = PT+1;
```

```

81  2      CALL RQ$EXIT$INTERRUPT(interrupt$level$3,
                        @status);

82  2      END HANDLER;

      /*****
      /* declarations for timer */
      *****/

83  1      DECLARE ADR2 LITERALLY '0D2H',
           /* address of timer 0 */
           ADR6 LITERALLY '0D6H',
           /* address for timer mode control word */
           MCR BYTE, /* mode control word */
           COUNTS BYTE; /* count for timer */

84  1      declare (n,m,hn,qn,lmax,i,ii,j,k,l,maxffc)
           integer;

85  1      declare (mc,maxmc) byte;
           /* use for analog data input */

86  1      declare index(513) integer;

87  1      declare areal(513) structure (mic(5) real);

88  1      declare aimag(513) structure (mic(5) real);

89  1      declare indata(1025) structure(mic(5)integer);

90  1      declare avchange(5) real;

91  1      declare change(5) real; /* average change and
           change used in detrending input data */

92  1      declare (angle,theta,fltn) real;

93  1      declare (sintable, costable) (513) real;

94  1      declare pi real;

95  1      declare ffc(120) real;

96  1      DECLARE HIGH LITERALLY '080H';
           /*port for high byte*/

97  1      declare(x) byte;

98  1      declare (pt, bpt) integer;

99  1      declare bptmax integer;

100 1      declare (sampledone, progdone) byte;

101 1      declare (woffset, roffset) integer; /* offsets

```



-47-

```
        for write and read locations in indata */
102  1  declare eekflag byte;      /* flag is 1 if
        sampling cycle finishes before program */
103  1  declare holdout(3073) integer;
104  1  declare (u, cos, sin) real;
105  1  declare bignum real;

106  1  call init$real$math$unit;
        /* initialize coprocessor */

        /*****
        /* initialize constants */
        *****/

107  1  pi=3.14159265;
108  1  n=512;
109  1  fltn=512.0;
110  1  hn = n/2;
111  1  qn = n/4;
112  1  m=9;
113  1  maxffc = 119;  /* maximum index for which
        ffc is non-zero */
        /* ffc's use symmetry in spectrum */
114  1  bignum = 32768.0;
```

-48-

```
115 1      call putstring
      (@(26,'How many fetal microphones '));
116 1      maxmc = low(unsign(getint)); /* maximum mc */
117 1      bptmax = 3073;

/* ***** */
/* initialize vector for bit reversal */
/* ***** */

118 1      initbitrev: do;
      /* Description of bit reversal accomplished in
      hardware. The bit reversal is accomplished by
      reversing the order of the separate address lines
      in bus 272 of dual ported ram 131. in the actual
      implementation, each entry a(i) is stored in
      memory location loc(i)=i-1. The bit reversal
      replaces loc(i) by rev(loc(i)), the binary
      integer obtained by writing loc(i) as an m place
      binary integer, including leading zeros, and
      reversing these digits to obtain rev(loc(i)).
      index(i)=rev(loc(i))+1. Bit reversal is its own
      inverse operation */

119 2      index(1)=1;
120 2      lmax=1;
121 2      ii=n/2;
122 2      loop1: do k=1 to m;
123 3          loop2: do l=1 to lmax;
124 4              index(l+lmax)=index(l)+ii;
125 4          end loop2;
126 3          ii=ii/2;
127 3          lmax=2*lmax;
128 3      end loop1;
```

```
129  2      end initbitrev;
      /* end routine to initialize bit reversal index */

      /******
      /*   initialize filter coefficients   /
      /******

130  1      do i=1 to 5;
131  2          ffc(i)=0.0;
132  2      end;

133  1      do i = 6 to 8;
134  2          ffc(i) = 1.0 - taper(5,9,i);
135  2      end;

136  1      do i = 9 to 81;
137  2          ffc(i) = 1.0;
138  2      end;

139  1      do i = 81 to 119;
140  2          ffc(i) = taper(80,120,i);
141  2      end;

      /* ***** */
      /* set up table of sines and cosines */
      /* ***** */
```

-50-

```

142  1      trigtangle: do i = 0 to n/8;
143  2          angle = 2.0*pi*float(i)/fltn;
144  2          u = angle*angle;
145  2          cos = 1.0-(u/2.0)*(1.0-(u/12.0)
              *(1.0-(u/30.0)*(1.0-u/56.0)));
146  2          sin = angle*(1.0-(u/6.0)*(1.0-(u/20.0)
              *(1.0-(u/42.0) *(1.0-u/72.0))));
147  2          costable(i) = cos;
148  2          sintable(i) = sin;
149  2          costable(qn-i) = sin;
150  2          sintable(qn-i) = cos;
151  2          costable(qn+i) = -sin;
152  2          sintable(qn+i) = cos;
153  2          costable(hn-i) = -cos;
154  2          sintable(hn-i) = sin;
155  2      end trigtangle;

```

```

/*****
/* assign interrupt procedure to interrupt level */
*****/

```

```

156  1      interrupt$task$flag = 0; /* no interrupt */
157  1      data$segment = 0;      /* task handler will
              load its own data segment */
158  1      interrupt$handler = INTERRUPT$PTR(HANDLER);
159  1      rerun;;
160  1      CALL RQ$SET$INTERRUPT (interrupt$level$3,
              interrupt$task$flag,
              interrupt$handler,
              data$segment,
              @status);

```

```
161  1      do i = 1 to 2*n;
162  2          do mc = 0 to maxmc;
163  3              indata(i).mic(mc) = 0;
164  3          end;
165  2      end;

166  1      PT = 1;
167  1      bpt = 1;
168  1      progdone = 0;
169  1      SAMPLEDONE = 0;
170  1      woffset = 0;
171  1      roffset = n;
172  1      eekflag = 0;

      /*****
      /* start timer when ready */
      *****/

173  1      MCR = 01100000B; /* 01 selects counter 1
                           10 selects read/write msb
                           000 selects mode 0 (interrupt)
                           0 selects hex timer */
174  1      COUNTS = 00AH;
175  1      OUTPUT(ADRS) = MCR;
      /*load mode control word into timer*/
```

```
176 1      OUTPUT(ADR2) = COUNTS; /* start timer */

177 1      X = INPUT(HIGH); /* read high byte
                        to set eoc bit and initialize adc */

/* **** */
/* TEMPORARY FOR TESTING SAMPLING CYCLE */
/* **** */

178 1      CALL PUTSTRING(@(14,'BEGIN SAMPLING'));

/* **** */
/* read in input data */
/* **** */

179 1      module1: do;

                /* **** */
                /* set flags */
                /* **** */

180 2      progdone = 0;
181 2      sampledone = 0;
182 2      eekflag = 0;

183 2      getdata: do;
                /* now detrend input data */
184 3          do mc = 0 to maxmc;
185 4              avchange(mc)=float(indata(n+roffset).mic(mc)
                        + indata(n+roffset-1).mic(mc)
                        + indata(n+roffset-2).mic(mc)
                        + indata(n+roffset-3).mic(mc)
                        - indata(roffset+4).mic(mc)
                        - indata(roffset+3).mic(mc)
                        - indata(roffset+2).mic(mc)
                        - indata(roffset+1).mic(mc))
                        / (4.0*(fltn-4.0));
```

-53-

```
186  4      change(mc)= avchange(mc);
187  4      do i = 2 + roffset to n + roffset;
188  5          indata(i).mic(mc) = indata(i).mic(mc)
              - fix(change(mc));
189  5          change(mc) = change(mc) + avchange(mc);
190  5      end;
191  4      end;

192  3      do i = 1 to n;
193  4          do mc = 0 to maxmc;
194  5              areal(index(i)).mic(mc) =
                  float(indata(i+roffset).mic(mc));
195  5              aimag(index(i)).mic(mc)=0.0;
196  5          end;
197  4      end;
198  3      end getdata;

199  2      hn=n/2;
      /*****/
200  2      call putstring (@(20,'some detrended data'));
201  2      do i = 1 to 4;
202  3          call putint(fix(areal(i).mic(0)));
203  3      end;
      /*****/
```

-54-

```

204  2  fft: do;
205  3      declare (c,s) real;
206  3      declare (io,p,hp) integer;
207  3      declare (t1,t2,s1,s2,v1,v2) real;
208  3      p=2;
209  3      hp=1;
210  3      loop3: do l=1 to m;
          /* ANGLE=0.0; theta=pi/float(hp); */
211  4      loop4: do j=1 to hp;
          /* c=mqercos(angle); angle = (j-1)*pi/hp */
          /* s=mqersin(angle); */

212  5          c = costable((j-1)*(n/p));
213  5          s = sintable((j-1)*(n/p));
          /* cos(angle) and sin(angle) are the
             (j-1)*n/(2*hp)th entries in respective
             tables in trig fcn rom of cos and sin
             for angles from 0 to (n/2-1)*2*pi/n in
             steps of 2*pi/n */
214  5      loop5: do i=j to n by p;
215  6          io=i+hp;
216  6      loop6: do mc=0 to maxmc;
217  7          t1 = areal(io).mic(mc)*c
                - aimag(io).mic(mc)*s;
218  7          t2 = areal(io).mic(mc)*s
                + aimag(io).mic(mc)*c;
219  7          areal(io).mic(mc)
                = areal(i).mic(mc)-t1;
220  7          aimag(io).mic(mc)
                = aimag(i).mic(mc)-t2;
221  7          areal(i).mic(mc)
                = areal(i).mic(mc)+t1;
222  7          aimag(i).mic(mc)
                = aimag(i).mic(mc)+t2;
223  7      end loop6;
224  6      end loop5;
          /* angle=angle+theta; */

```

SUBSTITUTE SHEET



-55-

```

225  5      end loop4;
226  4      hp = p;
227  4      p=p*2;
228  4      end loop3;
229  3      end fft;
      /* end of module to compute fft */
230  2      end module1;

231  1      module2: do;
      /* module to filter output from fft */
      /* illustrates block 321 of fig. 11 */
      /* Next lines illustrate optional filtering to
      remove noise in fetal heart spectra due to maternal
      heartbeat. Linear filter is illustrated.
      a wide choice of linear and nonlinear filters is
      possible. ffc stores fetal filter coefficients for
      simple linear filter. mfc stores maternal filter
      coefficients for simple linear filter */
      /* These variables are used for clarity */
      /* In the actual implementation multiplication by 0
      is replaced by assignment of 0 and multiplication
      by 1 is not performed */

```

?FL/M-86 COMPILER EFHMDATA

08/23/85 15:08:18 PAGE 9

```

232  2      declare (p, hp, io) integer;
233  2      declare (c, s) real;
234  2      declare ps(120) structure (mic(5) real);
      /* will need care when mc=0 is used for maternal */
235  2      declare (zcreal, acimag) (513) real;
236  2      declare snr(5) real;
237  2      declare (dp1, v1, v2, si, en, na, t1, t2) real;
238  2      declare ax(41) real, op(40) real;
239  2      declare (retrend, trend) real;
240  2      declare breal(513) real;

241  2      filter1: do;

```

```

242  3      do mc=0 to maxmc;
243  4          areal(1).mic(mc) = 0.0;
244  4          aimag(1).mic(mc) = 0.0;
245  4      end;

246  3      do i = 2 to maxffc;
247  4          do mc=0 to maxmc;
248  5              areal(i).mic(mc) = areal(i).mic(mc)*ffc(i);
249  5              aimag(i).mic(mc) = aimag(i).mic(mc)*ffc(i);
250  5              aimag(n+2-i).mic(mc)
                    = aimag(n+2-i).mic(mc) * ffc(i);
251  5              areal(n+2-i).mic(mc)
                    = areal(n+2-i).mic(mc) * ffc(i);
252  5          end;
253  4      end;

254  3      do i = maxffc+1 to n+1-maxffc;
255  4          do mc=0 to maxmc;
256  5              areal(i).mic(mc) = 0.0;
257  5              aimag(i).mic(mc) = 0.0;
258  5          end;
259  4      end;
260  3      END FILTER1;

261  2      FILTER: DO;
/* Filter fetal spectra of mic 0 through 3 by
forming their projections orthogonal to the
maternal spectrum, mic 4. If desired the amount of
filtering may be reduced by multiplying dpl by a
factor less than 1. dpl holds dot products (imag
part is zero)      */
/* calculate 1/2 square of norm of mic 4 omitting
constant term (i=1)      */

262  3      na=0.0;
263  3      do i=2 to maxffc;
264  4          na = na+areal(i).mic(0)*areal(i).mic(0)
                + aimag(i).mic(0) * aimag(i).mic(0);
265  4      end;

```

-57-

```

266   3      if na = 0.0 then go to quality;

268   3      do mc=1 to maxmc;
          /* calculate dot product of areal and aimag for
          all mic, omitting constant (first) entry */
269   4          dp1=0.0;
270   4          do i=2 to maxffc;
271   5              dp1 = dp1+areal(i).mic(mc)*areal(i).mic(0)
                  + aimag(i).mic(mc)*aimag(i).mic(0);
272   5          end;
273   4          dp1 = dp1/na;
          /* form orthogonal complement */
          /* remove maternal heartbeat (mic(0)) component*/
274   4          do i = 1 to maxffc;
275   5              areal(i).mic(mc) = areal(i).mic(mc)
                  - dp1 * areal(i).mic(0);
276   5              aimag(i).mic(mc) = aimag(i).mic(mc)
                  - dp1 * aimag(i).mic(0);
277   5          end;
278   4          end;
279   3      end filter;
          /* end of module to filter output from fft */

280   2      quality: do;
          /* module to evaluate quality of each filtered
          fetal heart spectrum. */
          /* illustrates block 316 of fig. 11 */
          /* ps holds the power spectrum */
          /* ps(i).mic(mc) holds the (i-1)st component,
          namely that of i-1 cycles per sampling period,
          of the power spectrum of the filtered signal
          of microphone i */

```

-58-

```
281  3      do mc = 1 to maxmc;
282  4          do i = 2 to maxffc;
283  5              ps(i).mic(mc) = areal(i).mic(mc)
                      * areal(i).mic(mc)
                      + aimag(i).mic(mc)*aimag(i).mic(mc);
284  5          end;
285  4      end;
```

/\* Each fetal heart signal energy is estimated by adding components of its filtered power spectrum representing a range of 1 to 4 hz (60-240 beats per minute). other ranges may be used. \*/  
/\* snr is the vector of signal to signal plus noise (total) energies for filtered fetal heart spectra.\*/

```
286  3      do mc = 1 to maxmc;
287  4          si = 0.0;
288  4          en = 0.0;
```

/\* respective estimated amplitude of signal and of total spectrum, omitting constant (first) entry in each case \*/

```
289  4          do i = 2 to 11;
290  5              en = en + ps(i).mic(mc);
291  5          end;

292  4          do i = 12 to 43;
293  5              si = si + ps(i).mic(mc);
294  5          end;

295  4          en = en + si;
```

-59-

```

296  4      do i = 44 to maxffc;
      /* can stop as soon as ps(i).mic(mc)=0      */
297  5          en = en + ps(i).mic(mc);
298  5      end;

299  4      if en = 0.0 then snr(mc) = 0.0;
301  4          else snr(mc) = si/en;
302  4      end;
303  3  end quality; /*end of module to evaluate
      quality of each filtered fetal
      heart spectrum */

304  2  combine: do i = 1 to n; /*module to calculate
      combined, filtered spectrum of fetal heart signals.*/
      /* illustrates block 317 of fig. 11.      */

305  3      areal(i) = 0.0;
306  3      acimag(i) = 0.0;

307  3      do mc = 1 to maxmc;
308  4          areal(i)=areal(i)+snr(mc)*areal(i).mic(mc);
309  4          acimag(i)=acimag(i)+snr(mc)*aimag(i).mic(mc);
310  4      end;
311  3  end combine;

      /* end of module to calculate combined, filtered
      spectrum of fetal heart signals      */

```

```
312  2  distribution: do;
      /* module to compute distribution of values
      of fetal heart beat. */
      /* illustrates block 317 of fig. 11. */
      /* that portion of the power spectrum representing
      heart rates of approximately 60-240 beats per
      minute is first computed and then smoothed */
      /* variables ax and op hold part of power spectrum
      and part of smoothed power spectrum, respectively.
      compute appropriate part of power spectrum of
      enhanced, composite fetal heart spectrum */
313  3      do i = 6 to 40; /* beats/minute = 60*i/8.53 */
314  4          ax(i) =
              acreal(i)*acreal(i)+acimag(i)*acimag(i);
315  4      end;
      /* smooth power spectrum, obtaining smoothed
      spectrum op(-). */
      /* simple linear smoothing illustrated */
316  3      do i = 7 to 39;
317  4          op(i) = (ax(i-1)+4.0*ax(i)+ax(i+1))/6.0;
318  4      end;
      /* Send results to interface 210 and then to
      output devices 211. Print here for illustration*/
319  3      call putstring
              (@(29,'smoothed fetal heart spectrum'));
320  3      do i = 9 to 39;
321  4          call putint(fix(60.0*float(i-1)/8.53));
322  4          op(i) = op(i)/bignum;
323  4          if op(i) >30000.0 then op(i) = 31000.0;
325  4          call putint(fix(op(i)));
326  4          call putskip;
```

-61-

```
327  4      end;
328  3      end distribution;
      /* end of module to compute distribution of
      values of fetal heart beat */

329  2      inverse: do;
      /* module to perform inverse fft. illustrates
      block 326 in fig. 11. */
      /* floating point arithmetic is used in this
      program unless fast floating point operations
      are available it is replaced by integer
      arithmetic using standard descriptions of
      floating point and fixed point arithmetic */
      /* inverse fft before bit reversal */
      /* b(-) holds output */
330  3          p = n;
331  3          hp = p/2;
332  3          do l = m to 1 by -1;
              /* angle = 0.0;
              theta = pi/float(hp); */
333  4              do j = 1 to hp;
                  /* c = mquercos(angle);
                  s = mqersin(angle); */

334  5                  c = costable((j-1)*(n/p));
335  5                  s = sintable((j-1)*(n/p));
                  /* angle = (j-1)*pi/hp. cos(angle) and
                  sin(angle) are the (j-1)*n/(2*hp) th
                  entries in respective tables in trig
                  fcn rom of cos and sin for angles from 0
                  to (n/2-1)*2*pi/n in steps of 2*pi/n */
```

SUBSTITUTE SHEET

```
336 5      do i = j to n by p;
337 6          io = i + hp;
338 6          t1 = acreal(i)-acreal(io);
339 6          t2 = acimag(i)-acimag(io);
340 6          acreal(i) = (acreal(i)+acreal(io))/2.0;
341 6          acimag(i) = (acimag(i)+acimag(io))/2.0;
342 6          acreal(io) = (t1*c + t2*s)/2.0;
343 6          acimag(io) = (-t1*s + t2*c)/2.0;
344 6      end;
          /* angle=angle+theta; */
345 5      end;
346 4      p = hp;
347 4      hp = hp/2;
348 4      end;
          /* Description of bit reversal accomplished in
          hardware. The bit reversal is accomplished by
          reversing the order of the separate address
          lines in bus 272 of dual-ported ram 168 */
349 3      do i = 1 to n;
350 4          breal(i) = acreal(index(i)); /* reals only */
351 4      end;
352 3      outdata: do;
          /* FIRST REVERSE THE PREVIOUS DETRENDING */
353 4          retrend = 0.0;
354 4          do mc = 1 to maxmc;
```



-63-

```
355 5      retrend=retrend+snr(mc)*avchange(mc);
356 5      end;
357 4      trend = retrend;
358 4      do i = 2 to n;
359 5          breal(i) = breal(i)+trend;
360 5          trend = trend+retrend;
361 5      end;
362 4      do i= 1 to n;
363 5          if abs(breal(i)) > 30000.0
              then breal(i) = -31000.0;
365 5          holdout(bpt) = fix(breal(i));
366 5          bpt = bpt + 1;
367 5      end;
368 4      end outdata;
369 3      end inverse;
              /* end of module to compute inverse fft */
370 2      end module2; /* end of program */

371 1      call putstring(@{1,'*'});

372 1      progdone = 1;
373 1      if eekflag = 1 then
              call putstring(@{16,'program too slow'});

375 1      do while sampledone = 0; /* wait */
376 2      end;

377 1      woffset = n-woffset;
378 1      roffset = n-roffset;

379 1      if bpt < bptmax + 1 - n then go to module1;
```

SUBSTITUTE SHEET

```
381 1 call rq$reset$interrupt(interrupt$level$3,  
                                @status);  
382 1 call putstring(@{4,'bpt '});  
383 1 call putint(bpt);  
384 1 call putskip;  
  
385 1 module3: do;  
386 2   declare i integer;  
387 2   call putstring(@{37,  
                      'type 1 to save on disk, 0 to continue'});  
388 2   i = getint;  
389 2   call putskip;  
390 2   if i=1 then call writefile(@holdout(n+1),2560);  
392 2 end;  
393 1 module4: do;  
394 2   call putstring (@{42,  
                      'type 1 to print output, return to continue'});  
395 2   i=getint;  
396 2   if i=1 then do;  
398 3     do i = n+1 to bpt;  
399 4       call putint(holdout(i));  
400 4       end;  
401 3   end;  
402 2   call putstring  
      (@{31,'type 1 to rerun, return to stop'});  
403 2   i = getint;  
404 2   if i = 1 then go to rerun;  
406 2 end module4;  
  
407 1 call endout;  
408 1 end efhmdata;
```

This illustrates the use of a variety of computer languages to implement the instant filtering algorithms. Several differences should be noted:

(1) The PL/M program demonstrates simultaneous data acquisition and processing, including the use of an analog to digital converter (the Intel iSBX 311). Data is gathered for five sampling cycles, and may then be displayed on a computer screen or saved on a disk. In the latter case, data on the fetal heartbeat itself, as opposed or in addition to variability data may be thus saved.

(2) Ordinary complex multiplication replaces fast complex multiplication due to the speed of multiplication on the 8087 coprocessor which makes the standard complex multiplication algorithm faster.

(3) Provision is made for varying the number of fetal microphones. This allows the simplicity of using one or usually a few microphones in the majority of cases where a relatively strong fetal heart signal can be obtained, and also the ability to add more microphones to adequately detect and process weak fetal heart signals or signals in the presence of more noise.

(4) Multiplications by 0 and storage of 0 entries are largely eliminated, saving time and space.

(5) Data is detrended before the Fast Fourier Transform and retrended after the inverse Fast Fourier Transform in order to eliminate any potential problems near the ends of each sampling cycle.

(6) Additional increases in speed are achieved through the use of symmetry properties of Fourier Transforms of real data.

A slight increase in speed may be obtained by

-66-

indexing arrays used in the Fourier transform and inverse Fourier transform from 0 to  $n-1$  rather than 1 to  $n$ .

The Burrus and Parks algorithm may be used to speed up the Fast Fourier Transform of real data by storing and processing  $n$  real data points in an array of  $n/2$  complex numbers. Their algorithm contained an obvious sign error. Here is an implementation of a corrected algorithm, followed by a similar but novel algorithm for the inverse Fast Fourier Transform.

-67-

/\*\*\* FOURIER TRANSFORM OF REAL DATA

```

106  2  fft: do;
107  3      declare (c,s) real;
108  3      declare (io,p,hp) integer;
109  3      declare (t1,t2,s1,s2,v1,v2) real;
110  3      p=2;
111  3      hp=1;
112  3      loop3: do l=1 to m-1;
113  4          loop4: do j=1 to hp;
114  5              c=costable(n*(j-1)/hp);
                  /* angle = (j-1)*pi/hp. */
115  5              s=sintable(n*(j-1)/hp);
                  /* cos(angle) and sin(angle) are the */
                  /* (j-1)*n/hp th entries in respective tables*/
                  /* in trig fcn rom of cosand sin for angles */
                  /* from 0 to pi insteps of pi/n */
116  5              loop5: do i=j to hn by p;
117  6                  io=i+hp;
                  /* the next loop performs fast complex */
                  /* multiplication */
                  /* if a fast multiply is available then it
                     should be by the ordinary */
                  /* implimentation of complex multiply */
118  6                  loop6: do mc=0 to maxmc;
119  7                      v1= areal(io).mic(mc)*c;
120  7                      v2= aimag(io).mic(mc)*s;
121  7                      t1= v1-v2;
122  7                      t2= (areal(io).mic(mc)
                              +aimag(io).mic(mc))*(c+s)
                              -v1-v2;
123  7                      areal(io).mic(mc)

```

-68-

```

                                = areal(i).mic(mc)-t1;
124   7      aimag(io).mic(mc)
                                = aimag(i).mic(mc)-t2;
125   7      areal(i).mic(mc)
                                = areal(i).mic(mc)+t1;
126   7      aimag(i).mic(mc)
                                = aimag(i).mic(mc)+t2;
127   7      end loop6;
128   6      end loop5;
129   5      end loop4;
130   4      hp = p;
131   4      p=p*2;
132   4      end loop3;

133   3      do mc = 0 to maxmc;
134   4          ureal(1).mic(mc)=areal(1).mic(mc);
135   4          uimag(1).mic(mc) = 0.0;
136   4          vreal(hn).mic(mc)
                                =aimag(1).mic(mc);
137   4          vimag(hn).mic(mc) = 0.0;
138   4          do i = 2 to hn;
139   5              ureal(i).mic(mc)
                                =(areal(i).mic(mc)
                                +areal(hn+2-i).mic(mc))/2.0;
140   5              uimag(i).mic(mc)
                                =(aimag(i).mic(mc)
                                -aimag(hn+2-i).mic(mc))/2.0;
141   5              vreal(hn+1-i).mic(mc)
                                = (aimag(i).mic(mc)
                                +aimag(hn+2-i).mic(mc))/2.0;
142   5              vimag(hn+1-i).mic(mc)
                                = (areal(hn+2-i).mic(mc)
                                - areal(i).mic(mc))/2.0;

/* this computes
-1/2*(areal(i).mic(mc)-areal(hn+2-i).mic(mc)*/

```

SUBSTITUTE SHEET

-69-

```
143  5      end;
144  4      do i = 1 to hn;
145  5          areal(i).mic(mc)
              =ureal(i).mic(mc)
              +vreal(hn+1-i).mic(mc)
              *costable(2*(i-1))
              -vimag(hn+1-i).mic(mc)
              *sintable(2*(i-1));
146  5      aimag(i).mic(mc)
              =uimag(i).mic(mc)
              +vimag(hn+1-i).mic(mc)
              * costable(2*(i-1))
              +vreal(hn+1-i).mic(mc)
              * sintable(2*(i-1));
147  5      end;
148  4      do i = hn+1 to n;
149  5          areal(i).mic(mc)
              =ureal(i-hn).mic(mc)
              +vreal(n+1-i).mic(mc)
              * costable(2*(i-1))
              -vimag(n+1-i).mic(mc)
              * sintable(2*(i-1));
150  5      aimag(i).mic(mc)
              =uimag(i-hn).mic(mc)
              +vimag(n+1-i).mic(mc)
              * costable(2*(i-1))
              +vreal(n+1-i).mic(mc)
              * sintable(2*(i-1));
151  5      end;
152  4      end;
153  3      end fft;
```

-70-

```
/** INVERSE FOURIER TRANSFORM OF REAL DATA **/  
267 2 inverse: do; /* b(-) holds output */  
268 3 do i = 1 to hn;  
269 4 ureal(i).mic(0) = (acreal(i)  
+ acreal (hn+2-i))/2.0;  
270 4 uimag(i).mic(0) = (acimag(i)  
- acimag (hn+2-i))/2.0;  
271 4 vreal(i).mic(0)  
= ((acreal(i)-acreal (hn+2-i))  
* costable(2*(i-1))  
+ (acimag(i)+acimag (hn+2-i))  
* sintable(2*(i-1))) /2.0;  
272 4 vimag(i).mic(0)  
= ((acimag(i)+acimag (hn+2-i))  
* costable(2*(i-1))  
- (acreal(i)-acreal (hn+2-i))  
* sintable(2*(i-1))) /2.0;  
273 4 end;  
  
274 3 do i = 1 to hn;  
275 4 acreal(i) = ureal(i).mic(0)  
- vimag(i).mic(0);  
276 4 acimag(i) = uimag(i).mic(0)  
+ vreal(i).mic(0);  
277 4 end;
```



```
278 3    p = hn;
279 3    hp = p/2;
280 3    do l = m-1 to 1 by -1;
281 4        do j = 1 to hp;
282 5            c = costable(n*(j-1)/hp);
283 5            s = sintable(n*(j-1)/hp);
284 5            do i = j to n by p;
285 6                io = i + hp;
286 6                t1 = acreal(i)-acreal(io);
287 6                t2 = acimag(i)-acimag(io);
288 6                acreal(i) = (acreal(i)
                               +acreal(io))/2.0;
289 6                acimag(i) = (acimag(i)
                               +acimag(io))/2.0;
290 6                acreal(io)=(t1*c+t2*s)/2.0;
291 6                acimag(io)=(-t1*s+t2*c)/2.0;
292 6            end;
293 5        end;
294 4        p = hp;
295 4        hp = hp/2;
296 4    end;

297 3    do i = 1 to hn;
298 4        breal(2*i-1) = acreal(index(i));
299 4        breal(2*i) = acimag(index(i));
300 4    end;
```

The choice of algorithm depends upon the tradeoff between space and time; since the above algorithms need more space.

Finally, in order to save space, variables in all programs may be overlaid when they are no longer needed.

The bit reversals are easily done in hardware by the addressing scheme of RAM 131. To do this for a sampling cycle of  $n = 2^m$  samples,  $m$ -bit addresses are used, and the digitized  $i$ th sample of each sampling cycle is stored in location  $rev(i-1)$  where for non-negative  $j < n$ ,  $rev(j)$  is the binary integer obtained by writing  $j$  as an  $m$ -bit binary integer, including leading zeros, and then reversing the order of the bits. Software controlled switches (not shown) may be used to vary  $m$  and  $n$  for various sampling cycles without requiring additional processing time.

A second microprocessor 155, e.g. type 80186, 80286, 68000, 68020, or 16032, reads the digitized microphone outputs (fetal heartbeat signals and optionally one maternal heartbeat signal) from dual-ported RAM units 131. This latter processor is run at a high clock speed, 8-16 MHz, in order to perform multiple (typically 4-5) FFTs, digital filtering, one inverse FFT, and some auxiliary computations within the time of one sampling cycle containing 128-8192 or more samples (typically 4.27-68.28 seconds in US versions, 5.12-82 seconds in European versions).

Optionally, microprocessor 155 may be supplemented by one or more of dedicated FFT chip 161, dedicated inverse FFT chip 162, arithmetic coprocessor, or another dedicated processor (the last two not shown).

Box 305 denotes the hardware associated with microprocessor 155. A machine language program for the FFT, later inverse FFT, and other computations is stored in ROM 156, and values of the appropriate trigonometric expressions

are stored in ROM 157. RAM 159 is used for computing. The final output from microprocessor 155 is sent to dual-ported RAM 168.

Various techniques are useful in obtaining the appropriate speed.

(1) Perform the bit reversals or input permutations used in the FFT in hardware, by the way RAM 131 is addressed, and perform analogous bit reversals used in a later inverse FFT by the way RAM 168 is addressed.

(2) Perform each step in the transform for all four signals consecutively or in parallel with additional chips or processors.

(3) Perform all multiplications as (typically 16 bit) integer multiplications, followed by suitable shifts on a microprocessor 155 such as the 80186 or 68000 with fast multiply instructions. A coprocessor, additional processors, and/or fast floating point multiplication instructions, where available, may also be used.

(4) Read precomputed values of required trigonometric functions from ROM instead of computing them at run time.

(5) Alternatively, compute the FFT with a dedicated special chip 161. In this case, the processor 155 controls the flow to and from chip 161. The separate chip gains processing speed at the expense of slight additional complexity. In this case, the inverse FFT, described below, could also be performed by a dedicated chip 162.

For a sampling cycle of 1024 samples, performing the FFT on four inputs, and the inverse FFT on the combined signal as described above requires 270,000-300,000 arithmetic operations, not including operations with indices, depending upon implementation. A maximum of 390,000 total operations

is required. Thus, after allowing for filtering and other processing, speeds of 60,000 operations per second could handle 1024 data points per 8.53 seconds, a sampling rate of 120 samples per minute. Other typical required speeds are as follows:

Samples per minute	No. of inputs	Samples per cycle	Length of sampling cycle in seconds	Speed in arith. ops. per second
3600	4	1024	17.07	30,000
3600	5	1024	17.07	35,000
7200	5	1024	8.53	70,000
7200	4	2048	17.07	70,000
7200	5	2048	17.07	80,000
7200	5	4096	34.14	90,000
14400	5	4096	17.07	180,000

A 68000 microprocessor, running at 10 MHz, performs 100,000 16-bit operations per second. The use of other possible chips such as the 80186, 80286, or 16032, coprocessors, or additional processors, depending on the required processing rate, will be clear to skilled artisans.

The output signals from the FFT each describe the frequencies present in one fetal (or maternal) heartbeat signal from one microphone. The fetal heartbeat frequency signals are first separately filtered and then added to produce a single enhanced signal representing the signals present in the fetal heartbeat. These steps are performed by microprocessor 155 and the hardware in box 305 of Fig. 10.

The first step in filtering consists of multiplying certain components by fractions whose denominators are powers of 2, etc, and deleting other components, particularly the 60 Hz component, which may be readily and completely eliminated because the sampling rate is an integral multiple of 60 Hz. Frequencies in the approximate range .2 to 8-15 Hz should be passed unattenuated to preserve information about the fetal heart rate. Lines 2000-2290 of the program above illustrate this process.

If desired, noise in the fetal heartbeat caused by the maternal heartbeat may be largely removed in a novel way by replacing the fetal heartbeat signals by their projections orthogonal to the maternal signal, or by reducing one or more frequency components present in the maternal signal, or by reducing one or more ranges of frequencies present in the maternal signal. This is illustrated by lines 2500-2900 in the program above.

Other linear and non-linear techniques are available, and other frequencies may be passed if desired.

Microprocessor 155 then computes the power spectrum, that is, the amount of energy at each frequency, for each separate fetal heart signal.

The quality of each signal is then evaluated. As illustrated by lines 3000 through 3900 of the above program, this may be done by first computing power spectrum of each signal. The quality is then measured by the ratio of energy in the range 1-4 Hz (other ranges may be used if desired), roughly the rate of the fetal heartbeat, to the total energy, as an analog of the signal to signal plus noise ratio. If desired, another measure of quality may be computed at this point. The filtered signals in the frequency domain are then combined into a single signal, for example by a weighted average, weighting each signal in proportion to their quality, as illustrated by lines 4000-4900 of the above program.

-76-

5  
This is done instead of adding the original signals in order to maximize signal quality, especially to prevent noise in one microphone from overshadowing useful signals from one or more other microphones. In the case of placental monitoring, where there are several sound sources, one may correct for the effects of phase differences between the signals by another program if desired (not likely to be needed for fetal heart monitoring). One way to correct for phase differences is to dynamically modify the addresses used to write fetal heart signal samples into RAM 131.

Alternately, the fetal heartbeat signals could be added directly, weighting each signal in proportion to its average amplitude if desired, and the combined signal could then be filtered with the FFT, digital filters, and inverse FFT as above. The filtering techniques could also be applied to auxiliary input signals.

Next the portion of the power spectrum of the filtered, combined fetal heart spectra (frequency domain signals) which represents frequencies in the approximate range 1-4 Hz, or 60-240 beats per minute, computed and smoothed as illustrated by lines 5000-5900 of the above program. The resulting digital signal obtained by microprocessor 155 is sent to an interface 210 and then to the digital frequency output display unit 211 (part of 18). (Analog devices with D/A converters may be used instead if desired.)

In addition, the inverse FFT is applied to the resulting combined, filtered signal to produce an enhanced, filtered signal of the fetal heartbeat in the time domain in short segments, typically 4.27-82 seconds. A novel implementation of the inverse FFT is illustrated by lines 6000-6900 of the above program, which is specially designed so that the bit reversals form the final step which may again be done in hardware. This computation, except for the bit

reversals beginning at line 6280 if desired, may be done either directly by the processor or by the inverse FFT chip 162.

Final output from this processor 155 is stored in another dual-ported RAM 168. Note that this RAM is distinct from RAM 159 used by the processor for its own calculations. Also, ports for this RAM are switched at suitable intervals (typically 4.27-82 seconds) by the processor 145.

A third and final microprocessor 170 reads the RAM 168, using an address bus with the order of separate lines reversed as in RAM 131 to effect the bit reversals and sends the output at the proper time intervals to digital/analog convertor 181. Here there is a single signal representing a filtered combination of the multiple (typically 4) signals from microphones used to detect fetal heartbeat. Again, the microprocessor may control addressing without having data pass through the microprocessor. The output of the D/A converters is sent to respective linear amplifier 183 for scaling and isolation purposes. The output of this amplifier is the filtered, enhanced fetal heartbeat signal at a slight lag (two sampling cycles, or typically 8-164 seconds in the proposed embodiment). This signal is displayed on the fetal heart output display unit 209, part of 18. If desired for further processing such as computation of instantaneous heart rates within the processing unit or output devices, a similar digital output may be provided by sending data from the dual-ported RAM 168 to a suitable interface (not shown) at the sampling rate under the control of microprocessor 170.

Similarly, the processor 170 also reads the RAM units (not shown) associated with the other inputs and sends these signals to additional D/A convertors (not shown), linear amplifiers (not shown), and output units (part of 18). The RAM associated with auxiliary inputs is configured so that these signals have the same lag as fetal heartbeat.

Alternatively, a switch (not shown) bypasses the circuitry used to delay the auxiliary inputs (analog data acquisition unit, RAM, D/A converter, etc.) for synchrony with the delayed fetal heartbeat to allow undelayed signals from the auxiliary inputs. For example, in difficult labors, signals from the monitors of contractions, with no delay, could be used to initiate controlled breathing or regulate medicine dosage. Analog or digital outputs associated with auxiliary inputs are easily provided for displayed on respective display units (part of 18).

#### Outputs

First, the output from the amplifier 183 consists of an enhanced, filtered (analog) signal of the fetal heartbeat, presented at a short lag behind real time. This signal may be further processed (not shown) to compute instantaneous heart rates, beat to beat variability, smoothed beat to beat variability with an algorithm such as the moving average, and may also be processed with autocorrelation techniques. Alternatively, if desired to facilitate processing, a similar digital output may be obtained using a suitable interface (not shown) attached to the output of the D/A converter 181.

Auxiliary signals, such as data on the onset of contractions or intrauterine pressure, are delayed so as to be synchronized with the fetal heartbeat, and presented on the output unit 18. Thus acceleration, deceleration and variability may be correlated with the onset and duration of contractions or other data. Analog or digital versions may be provided.

The graphs in Fig. 14 of the frequency output on device 211 show the energy of the different fundamental frequencies or inverse heartrates, which make up the filtered, enhanced fetal heartbeat signal. Two idealized prototype enhanced power spectra to illustrate how variability of fetal heart rate during a sampling cycle is represented in the power



-79-

spectrum of that cycle. The fetal heart rate is displayed along the horizontal axis and relative energy (or optionally amplitude, or other measure of the size of Fourier coefficients) is displayed along the vertical axis. Normally, the measure of the size of coefficients is a fixed attribute for the monitor. The average heartrate of 140 in graph 401 is read from the horizontal coordinate 403 of the peak 404. Variability corresponds to the width 406 of the spectrum shown at about 15 beats per minute. Graph 402 represents the same average heartrate but a smaller variability, about 5 beats per second. This data is updated slightly before the other outputs are obtained, but persists or may be easily held until they are obtained.

All outputs may be represented on devices such as oscilloscopes, recording devices, or other standard output devices familiar to those skilled in the art. Further processing may also be performed at these points if desired.

In addition, properly synchronized records combining delayed fetal heart signals, and non-delayed auxiliary signals may be made by using modified strip chart recorders.

-80-

### Design of dual-ported RAM

The major components of dual-ported RAM units 131 and 168, except for controllers and interfaces, are shown in Figs 10 and 11. Except as noted, the RAM used to store auxiliary inputs is similar and not shown. Each RAM unit contains two sets of memory chips, 251 and 256. These chips are configured appropriately to handle 12-16 bit data words and, in the case of RAM 131 and 168, at least one word per sample in each sampling period (typically 512-4096 or more words). This is well within the range of sets of 16K or 64K RAM chips. The RAM used to delay auxiliary inputs must be twice as big in order to handle the required time delay of two sampling cycles.

There are also two ports 266 and 271, respectively. Finally there is a software controlled (via control 276) digital switch 261, which either connects ports 266 and 271 to chips 251 and 256 respectively, or vice versa. The connections are made via address lines 252, 257, 267, 272, data lines 253, 258, 268, 273, and control lines 254, 259, 269, 274.

In addition, in RAMs 131 and 168, but not the RAMs (not shown) used to hold auxiliary inputs, the address line 272 is inverted so that the bits in each address are reversed to perform the bit reversal or permutation of input in the FFT in hardware.

### Operation of dual-ported RAM

In RAM 131, microprocessor 145 is connected to switch control 276 and to port 266 for addresses and control, interface 129 is connected to port 266 for data, and microprocessor 155 is connected to port 271 for addresses, data, and control. (Alternatively, if desired, data from

interface 129 may pass through microprocessor 145.)

During the first sampling cycle, typically 17.07 seconds (this value will be used in subsequent discussion), port 266 is connected to memory chips 251. Thus data from the interface 129 is written into memory chips 251. At the same time, port 271 is connected to memory chips 256, but remains idle. At the end of this interval memory chips 251 contain the digitized fetal heartbeat signals obtained in the first sampling period. The signals from each auxiliary input are stored in the corresponding chips 251 in corresponding RAM units (not shown).

At the end of this sampling period, the digital switch 261 in RAM 131 is inverted under the control of microprocessor 145 via switch control 276. During the second sampling period, port 266 is connected to memory chips 256, which store digitized fetal heartbeat signals as described above. At the same time, port 271 is connected to memory chips 251. The signals in these chips 251 are read rapidly by microprocessor 155 which performs the FFT algorithm upon this data. In addition, by the way RAM 131 is addressed, the first part of the FFT is performed in hardware, requiring no execution time.

At the end of the second period, the digital switch 261 is again reversed under the control of microprocessor 145. As in the first interval, fetal heartbeat data is read into chips 251 from the interface, while, by analogy with the second period, microprocessor 155 reads chips 256 and applies the FFT to their data. At the end of each subsequent sampling period, connections and the roles of chips 251 and 256 are similarly reversed.

RAM 168 of Fig. 13 (only shown where different from Fig. 12) operates similarly. Microprocessor 155 is connected to port 266 for data, addresses and control; microprocessor 170 is connected to port 271 for addresses and control, D/A converter 181 and a (digital) interface (not shown) are

connected to port 271 for data, and microprocessor 145 is connected to switch control 276. Microprocessor 155 writes the filtered, enhanced digitized fetal heartbeat data into this RAM at the conclusion of its processing. Microprocessor 170 reads data, representing the enhanced fetal heart signal from this RAM into D/A converter 181 sequentially at the original sampling rate to generate the output. As above, microprocessor 145, possibly in combination with microprocessor 155, reverses the connections in the switch at the end of each sampling period. Thus data can be rapidly written to RAM 168 after processing, while other chips in RAM 168 are read at the sampling rate to generate the output. The required connections by sampling periods are:

Periods 1, 3, etc.	Port 266 to chips 251
	Port 271 to chips 256,
Periods 2, 4, etc.	Port 266 to chips 256
	Port 271 to chips 251.

The final bit reversal in the inverse FFT is performed by the addressing of RAM 168, requiring no execution time.

The dual-ported RAM associated with the auxiliary inputs (not shown) operates similarly, with the exception that connections are reversed after each pair of sampling periods. In this way, this RAM delays signals from the auxiliary inputs by the same amount (2 cycles or 34.14 seconds) as the sum of the delays in the fetal heartbeat RAMs 131 and 168. Note that these delays allow for the required processing time, whereas signals from the auxiliary inputs are not processed. Switching is analogous to that of RAM 131 except for the larger interval between reversals: for example, during sampling periods 1 and 2, port 266 is connected to chips 251.

There are other dual ported RAM units or control methods which will be clear to those skilled in the art.

### Logical processes

Referring to Figs. 8 and 9, and the program above, the logic of the processing of fetal heart signals (and maternal heart signal optionally used in the filtering process) 94 will now be described. Optionally, auxiliary signals may be similarly processed.

If desired, signals 94 may be optionally demodulated before further processing (not shown) to emphasize the frequencies in the fetal heartbeat. In this case, if the maternal heartbeat is to be used in filtering the fetal heartbeat, the maternal heartbeat is similarly demodulated (not shown).

The signals are then sent to analog data acquisition units 301, controlled by microprocessor 145. The units 301 also contain passive or active audio filters to suppress frequency components in excess of the sampling rate. This yields a less noisy sample and avoids aliases and ringing. The analog data acquisition units convert the analog heartbeat signals to digital signals in real time at the specified sampling rate, in the form of 12 to 16 bit wide data.

These signals are stored in dual ported RAM 131 for temporary storage before processing under the control of microprocessor 145.

At the end of each sampling cycle, the fetal heart signals previously stored in the dual-ported RAM 131 are processed. The first step 312 consists of a module to compute the FFT of each fetal heart signal (and a maternal heart signal if needed for later filtering), and is illustrated by lines 1000-1900 of the program above (the indicated bit reversals in lines 1200-1400 are performed in addressing RAM 131). Output from the FFT consists of the range of frequencies and amount of energy at each frequency present in each fetal heart signal. At a typical sampling rate of 7200

samples per minute, and a sampling period of 2048 samples or about 17.07 seconds, frequencies from .1 Hz through 60 Hz are represented. The outputs from the FFT are called frequency domain fetal heart signals.

The FFT is executed by microprocessor 155, possibly with a special purpose chip 161. All of the functions performed by microprocessor 155 are enclosed in a box 306, Fig. 11, using hardware enclosed in a box 305, Fig. 10.

Note that multiple (typically 4 or 5) signals are processed at 312 and elsewhere in Fig. 11. A single high-speed microprocessor (typically 8-16 MHz, 16 or 32 bits) may be used, optionally in conjunction with additional processors, coprocessors, or dedicated chips.

Each frequency domain fetal heartbeat signal is filtered by a module to filter spectra 321, illustrated by lines 2000-2900 of the above program, and executed by Microprocessor 155. The spectra first pass through a linear (illustrated by lines 2000-2290) or non-linear filter chosen to pass unattenuated frequency data low frequency data ranging from about .2 to 8-15 Hz, to attenuate higher frequencies as required for noise suppression including suppression of some of the maternal heartbeat signal, and, because of choice of sampling rate, to totally eliminate 60 cycle hum. The filtering process may be easily adapted to pass other frequencies. Next any noise in the fetal spectra due to the maternal heartbeat is effectively eliminated by replacing the fetal spectra by their projections orthogonal to the maternal spectra, as illustrated in lines 2500-2900. A dot product in which the constant term in each spectrum has weight zero is preferable and illustrated.

Next, as illustrated by lines 3000-3900, the quality of each filtered fetal spectrum is evaluated by a module 316. The power spectra are used to compute a measure of the relative quality of each fetal heart frequency domain signal.

One measure of the quality is the ratio between the amount of energy in the range 60-240 beats per minute (1 to 4 Hz) to the total energy, a rough measure of the signal to noise ratio.

Next the weighted sum of the filtered fetal heart frequency domain signals from 321, weighted according to signal quality, is computed by a module 323, illustrated by lines 4000-4900 of the above program.

As illustrated by lines 5000-5900 of the above program, the portion of the power spectrum of the combined filtered fetal spectra in the range of 60 to 240 beats per minute, is computed and smoothed at module 318 as a measure of the distribution of fetal heart rates. The resulting signal is then sent to an interface 210 and frequency output devices 211.

The combined filtered fetal spectrum is processed by the inverse FFT 326, as illustrated by lines 6000-6900 of the above program, to obtain an enhanced, filtered digital fetal heart signal. This signal is written into dual ported RAM 168, in such a way, if desired, as to effect the bit reversals illustrated by lines 6280-6330. The enhanced, filtered digital fetal heart signal from RAM 168 is read into a digital to analog (D/A) converter 181 at the sampling rate, under the control of microprocessor 170, and then amplified at 183. This yields a filtered, enhanced fetal heart signal (analog) in real time except for a delay of two sampling cycles, between 8 and 164 seconds, and typically about 34 seconds. The signal is then sent to suitable output devices 209, and further processed (not shown) to obtain data such as instantaneous heart rates. A similar digital signal may also be provided if desired.

The smoothed, combined power spectrum from module 318 is sent to a buffer 210 to delay it until it is synchronized with the fetal heart signal, and then displayed on a suitable digitally driven frequency output display 211. Again, this

signal lags behind the fetal heart rate by two sampling cycles.

Finally, signals from auxiliary inputs may be either displayed as received or delayed to be in synchrony with the delayed fetal heart signal, depending on a switch position. The delay is accomplished with dual-ported RAMs analogous to RAM 131 (except that (1) no address line is reversed and (2) twice as much storage is required for a two sampling cycle delay), and D/A convertors and amplifiers such as 181 and 183. The delayed signal allows comparison between contractions and acceleration, deceleration and variability. Real time tonometric data may be used to alert the mother to the onset of each contraction.

The logical flow in an alternate embodiment in which signals are combined in the time domain is similar except that a different measure of signal quality must be used. Details will be apparent from the above descriptions.

The sampling cycle is selected in order to satisfy one or more of the following goals. A short cycle minimizes the delay, and provides variability data for short time periods. A longer cycles may be used to improve signal quality. Cycles lasting from 8 to 41 seconds seem feasible. In all cases the length of a cycle must be a power of two times the interval between samples, and the latter must be harmonica related to 60 Hz in the US and 50 Hz in Europe in order to minimize hum.

Sampling rates should be as high as practical, with rates of 7200 samples per minute practical using microprocessors such as the 68000, and 240 samples per second or higher using the 68020.

Placental monitoring may be facilitated by using components of the spectrum of each input to regulate phasing of the inputs by adjusting the addresses used to write



digitized input signals into dual-ported RAM 131 to achieve the requisite time delays, and by demodulation if desired.

These techniques may also be used to enhance signals from auxiliary inputs or ultrasound monitors if desired.

#### Summary

A fetal monitor which makes essential use of telemetry and a digital signal processing technique called the FFT. The use of FFT techniques and addition of signals in the frequency domain, together with digital frequency domain filtering described above, yields an accurate fetal heart signal as well as an accurate measure of fetal heart rate (after further processing) by filtering out random noise and noise at irrelevant frequencies. These signals presented together with data on the onset and duration of contractions, yielding a good indication of timing and duration of deceleration during contractions. This permits accurate fetal monitoring without restricting the mobility of the mother in labor. Finally, by presenting spectral (frequency) data, we obtain the range of fetal heartrates present during the last sampling period, useful information about variability in fetal heartrate, a frequently critical diagnostic indicator.

In addition, this device can be used to monitor placental blood flow, or enhance ultrasound data, should that prove desirable.

## I CLAIM:

1. In a fetal monitor system for detecting electrical signals directly and/or indirectly generated by fetal heart or placental physiological activity, transmitting said detected signals to processing units, and enhancing and filtering said detected signals in said processing units, the improvement comprising transforming said detected signals to the frequency domain with the Fourier Transform techniques and performing said enhancing and filtering in the frequency domain.
2. The system of Claim 1 wherein said Fourier Transform techniques include the Fast Fourier Transform algorithm.
3. The system of Claim 1 wherein said frequency domain signals are used to measure variability.
4. The system of Claim 1 wherein said enhanced filtered signal is further enhanced by moving averages.
5. The system of Claim 1 wherein instantaneous rates and moving averages of said rates of said enhanced filtered signal are computed.
6. The system of Claim 1 wherein the frequency domain signals resulting from the Fourier Transform are filtered.
7. The system of Claim 6 wherein said sampling rates is harmonically related to local alternating current frequency so as to filter out noise from said current.
8. The system of Claim 1 wherein the Fourier Transform of maternal heart signals is also computed and the noise thereof in fetal heart signals is removed by filtering in the frequency domain.
9. The system of Claim 8 wherein said filtering includes projecting said fetal heart signals orthogonally to said maternal heart signal.

10. The system of Claim 1 wherein signals other than fetal heart or placental signals are auxiliary thereto, are not enhanced and processed but are delayed to be presented with said enhanced and processed fetal heart signal.
11. The system of Claim 10 wherein said delay is accomplished by temporary storage in dual-ported random access memory units.
12. The system of Claim 1 wherein said electrical signals include signals generated by sound or ultrasound from the fetal heart or placenta or by fetal heart electrical activity.
13. The system of Claim 12 wherein said electrical signals are generated by fetal heart sounds.
14. The system of Claim 13 including a plurality of detectors.
15. The system of Claim 14 wherein said detectors are four spaced microphones attached to the exterior maternal abdominal wall.
16. The system of Claim 12 wherein signals are sampled and digitized at regular intervals by analog to digital converters.
17. The system of Claim 16 wherein said Fourier Transform and subsequent filtering is applied sequentially to a repetitive sequence of sampling cycles, each consisting of a number of samples of said signals and said enhanced filtered signal is produced within two said sampling cycles.
18. The system of Claim 16 wherein the sampling rate is at least 3600 samples per minute.
19. The system of Claim 16 wherein the sampling rate is harmonically related to the local alternating current frequency.

20. The system of Claim 16 wherein said Fourier Transform and subsequent filtering is applied sequentially to a repetitive sequence of sampling cycles, each consisting of a number of samples of said signals, and said sampling cycles are chosen to encompass parts of contraction.

21. The system of Claim 16 wherein said Fourier Transform and subsequent filtering is applied sequentially to a repetitive sequence of sampling cycles, each consisting of a number of samples of said signals, and said sampling cycles are chosen to encompass whole contractions.

22. The system of Claim 16 wherein said Fourier Transform and subsequent filtering is applied sequentially to a repetitive sequence of sampling cycles, each consisting of a number of samples of said signals, and said sampling cycles are chosen to start at a predetermined point of contractions.

23. The system of Claim 1 wherein inverse Fourier Transform is applied subsequent to said Fourier Transform to recover time domain signals from said enhanced filtered frequency domain signals.

24. The system of Claim 23 wherein microprocessors are used to effect the Fourier Transform, filtering, and inverse Fourier Transform.

25. The system of Claim 24 wherein data is temporarily stored in dual-ported random access memory units.

26. The system of Claim 25 wherein said Fourier Transform is executed with the Fast Fourier Transform algorithm, and bit reversals of said Fast Fourier Transform are performed using reversed address lines in an address bus to said dual-ported random access memory units.

27. The system of Claim 25 wherein said inverse Fourier Transform is executed with the inverse Fast Fourier Transform algorithm, and bit reversals of said inverse Fast Fourier Transform are performed using reversed address lines in an address bus to said dual-ported random access memory units.

28. The system of Claim 1 including a plurality of detectors for generating said fetal heart or placental signals.

29. The system of Claim 28 wherein the Fourier Transform is applied separately to each of said detected fetal heart or placental signals to obtain corresponding separate frequency domain signals which are combined to form a single composite frequency domain fetal heart signal and the inverse Fourier Transform is subsequently applied to this signal to recover a composite time domain signal.

30. The system of Claim 29 wherein said combination and subsequent inverse Fourier Transform is effected after filtering in the frequency domain to obtain enhanced composite frequency domain and time domain signals.

31. The system of Claim 29 wherein the combination of separate signals may change their original ratio in the ultimate signal.

32. The system of Claim 29 wherein said frequency domain fetal heart or placental data is used to measure variability.

33. The system of Claim 32 wherein said frequency domain data is smoothed and then used to measure variability.

34. The system of Claim 29 wherein said combination is a weighted average with weights determined by and directly related to computer estimation of the respective signal to noise ratios.

35. The system of Claim 34 wherein said respective estimates are directly related to the fraction of respective total energies which lie in certain frequency ranges.

36. The system of Claim 35 wherein said frequency range is 1 to 4 Hz.

37. In a system for detecting electrical signals generated directly or indirectly by heart physiological activity, the improvement comprising combining a plurality of said signals using a weighted average in which weights may vary through time as determined automatically by a microprocessor.

38. The system of Claim 37 wherein said weights are determined by and directly related to computer estimation of the respective signal to noise ratios.

39. The system of Claim 38 wherein said respective estimates are directly related to the fraction of respective total energies which lie in certain frequency ranges.

40. A fetal monitor method comprising detecting by a plurality of spaced sensors on the external maternal abdominal wall electrical signals directly and/or indirectly generated by fetal and/or maternal physiological activity, multiplexing said detected signals, transmitting wirelessly said multiplexed signals to a receiver, demultiplexing said transmitted signals to recover said detected signals, transforming said demultiplexed signals to the frequency domain with Fourier Transform techniques, enhancing, filtering, and combining said frequency domain signals, and applying the inverse Fourier Transform to said enhanced and filtered signals to recover time domain signals.

41. A fetal monitoring apparatus comprising a plurality of sensors to detect electrical signals directly and/or indirectly generated by fetal and/or maternal physiological activity, a multiplexer to combine said detected signals into a single multiplexed signal, a wireless transmitter to transmit said

multiplexed signal to a receiver, a demultiplexer to separate said received signal and recover said detected signals, an analog data acquisition unit to sample and digitize said recovered signals to obtain corresponding digital signals, microprocessors to (1) apply the Fourier Transform to said digital signals to obtain corresponding frequency domain signals, (2) enhance, filter, and combine said frequency domain signals, and (3) apply the inverse Fourier Transform to said combined frequency domain signals to recover enhanced and filtered time domain signals, and display devices to display said frequency and time domain signals.

42. In a fetal monitor apparatus including means for detecting electrical signals indirectly generated by fetal and/or maternal physiological activity, the improvement comprising a portable wireless transmitter attached to or worn by the mother for transmission of said detected signals to a receiver and means to filter and process said detected signals.

43. In a fetal monitor apparatus including means for detecting electrical signals directly and/or indirectly generated by fetal heart or placental physiological activity, transmitting said detected signals to processing units, enhancing and filtering said transmitted signals in said processing units, and outputting said enhanced and filtered signals in display, recording, control, and/or warning devices, the improvement consisting of analog data acquisition units to sample and transform said transmitted signals into digital time domain signals, and microprocessors to transform said digital signals to frequency domain by the Fourier Transform, and enhancing and filtering, and/or smoothing said frequency domain signal.

44. The apparatus of Claim 43 wherein said analog data acquisition unit includes analog filters for preconditioning said detected signals before sampling and digitizing.

45. The apparatus of Claim 43 including means to detect maternal uterine activity and detect and process fetal heart or placental

signals in a repetitive sequence <sup>-94-</sup> of sampling cycles each starting at a predetermined point in a contraction.

46. The apparatus of Claim 43 including dual-ported random access memory units for temporary data storage of digital time domain signals or intermediate digital data generated by said microprocessors.

47. The apparatus of Claim 43 wherein the execution of said Fourier Transforms is accelerated by fast multiply instructions, coprocessors, plural microprocessors each executing part of said Transforms, memory units sorting precomputed values of trigonometric functions, and/or dedicated Fourier Transform chips.

48. The apparatus of Claim 43 including additional detectors for electrical signals generated directly and/or indirectly by maternal heart physiological activity, analog to digital convertors to acquire and convert said maternal electrical signals to digital time domain maternal heart signals, microprocessors to (1) transform said digital time domain maternal heart signal to the frequency domain using the Fourier Transform, and (2) remove maternal heart noise from said fetal heart or placental frequency domain signals using said frequency domain maternal heart signal.

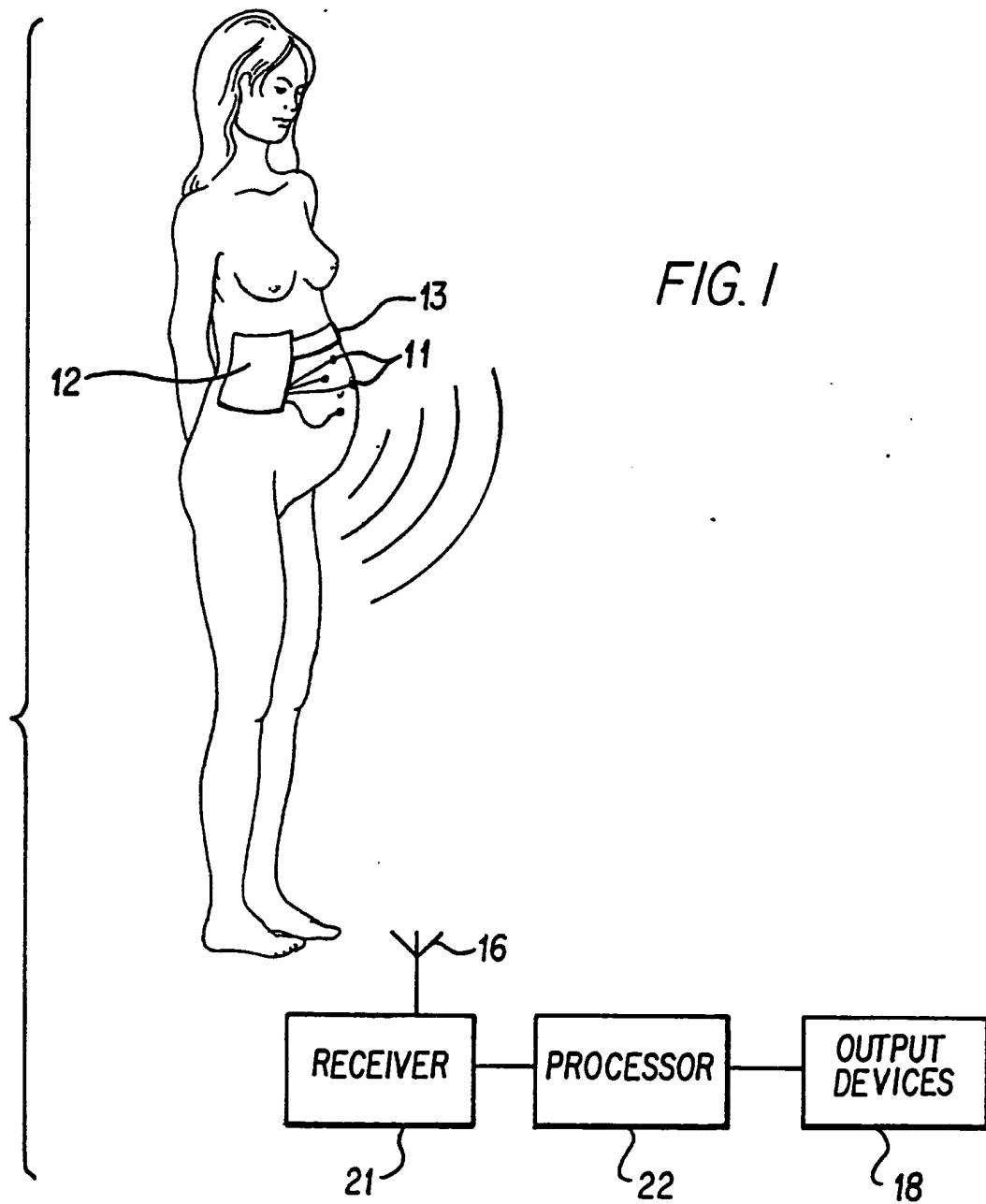
49. In apparatus for detecting electrical signals directly and/or indirectly generated by heart physiological activity, the improvement comprising a means to form a weighted average of a plurality of such signals in which the weights are computed by microprocessors and may vary through time.

50. The apparatus of Claim 49 in which such means comprise analog data acquisition units, microprocessors, and programs.

51. A fetal monitor microphone adapted to be positioned on a maternal abdomen for wireless transmission of detected fetal and/or maternal physiological activity comprising a transducer to detect signals from said activity and a filter to process said signals before said transmission.

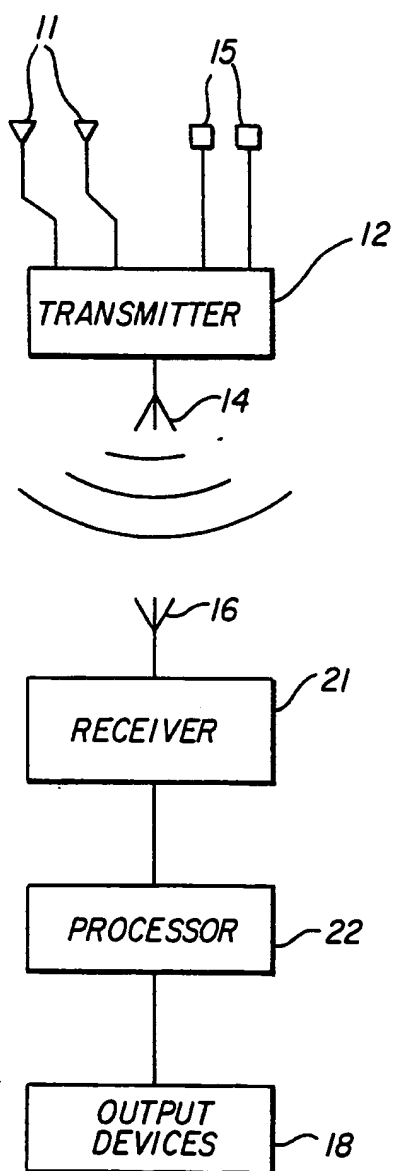


1/10



2 / 10

FIG. 2



3/10

FIG. 3

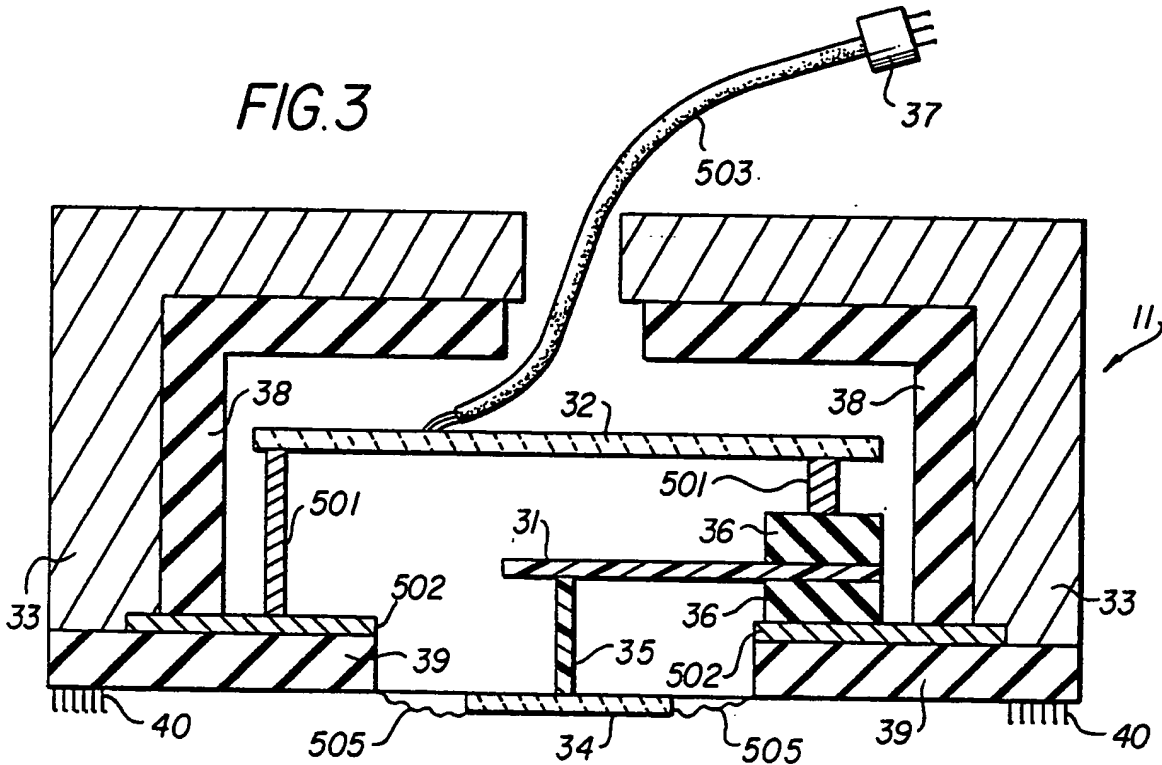
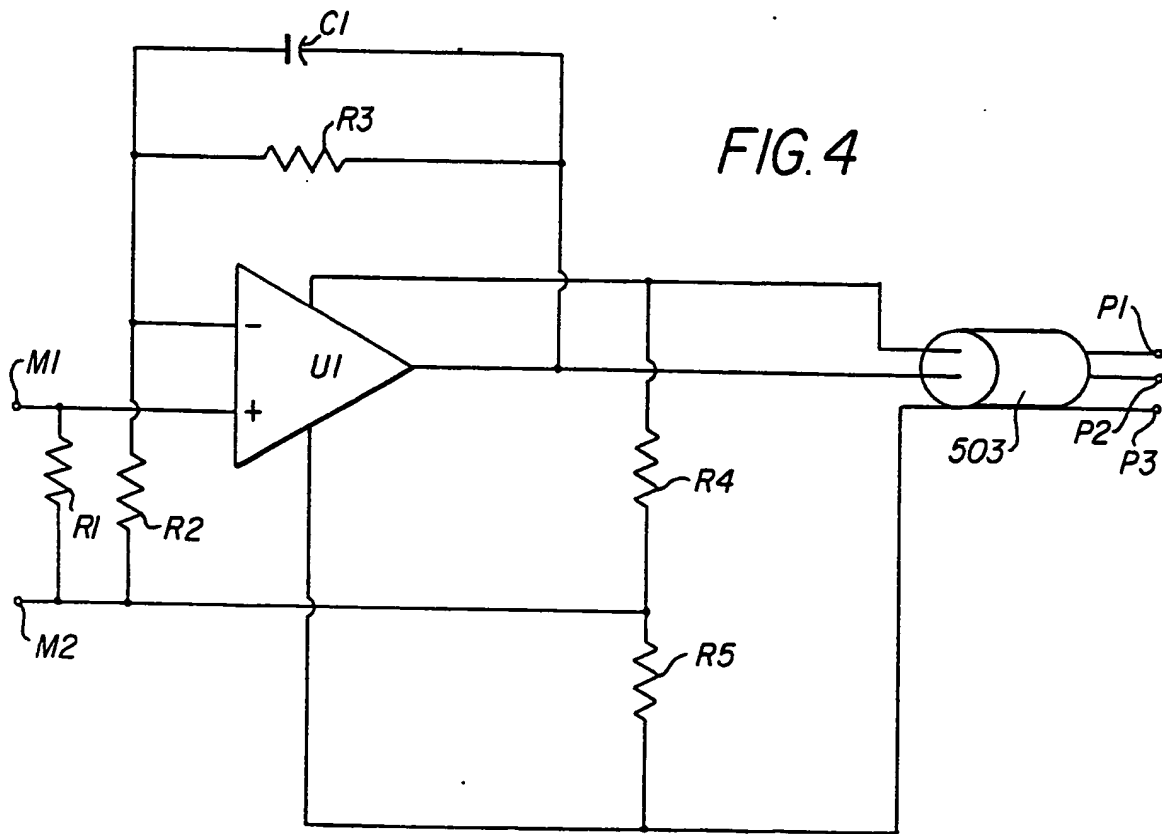


FIG. 4





5/10

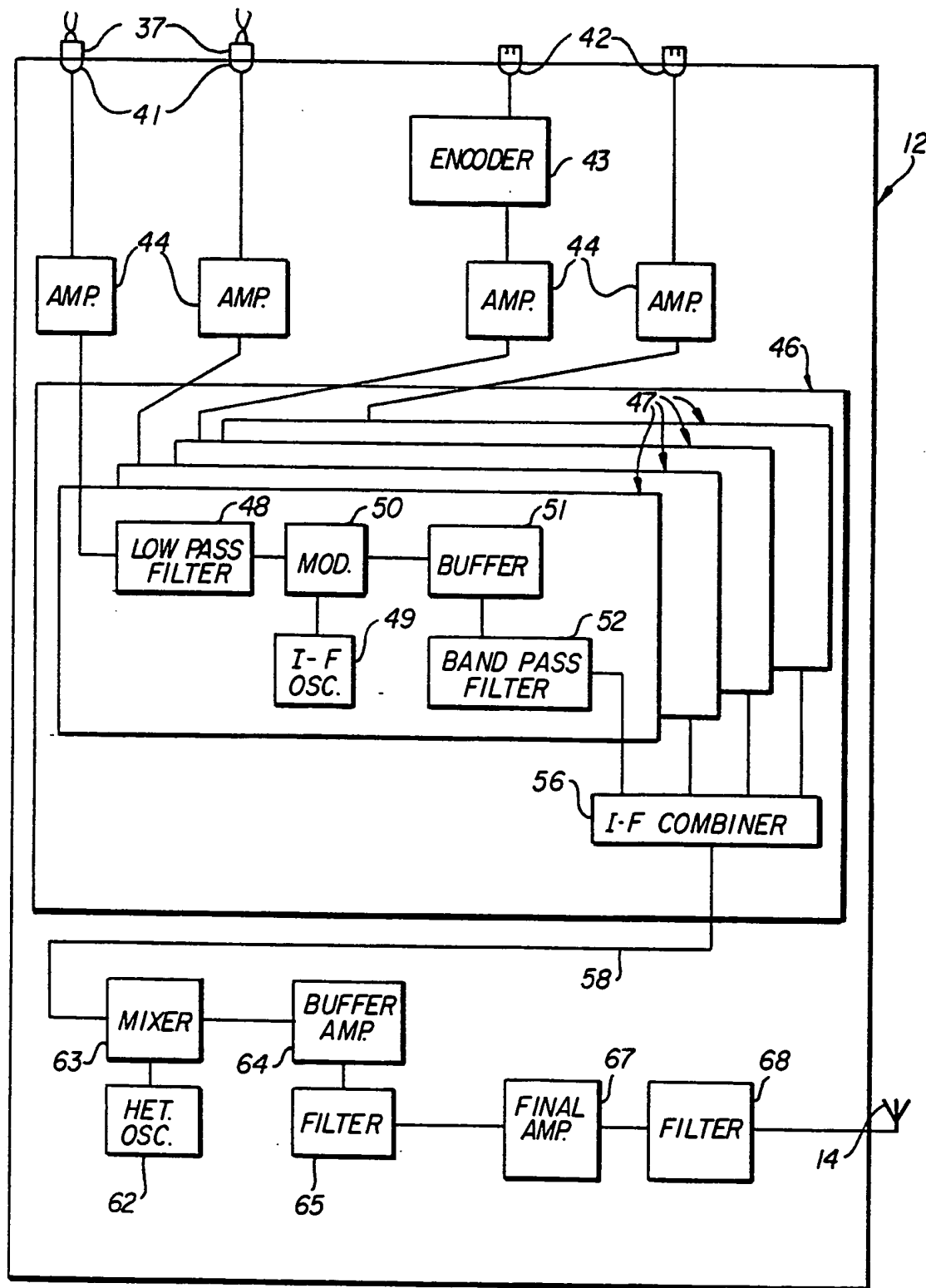


FIG. 6

6/10

FIG. 7

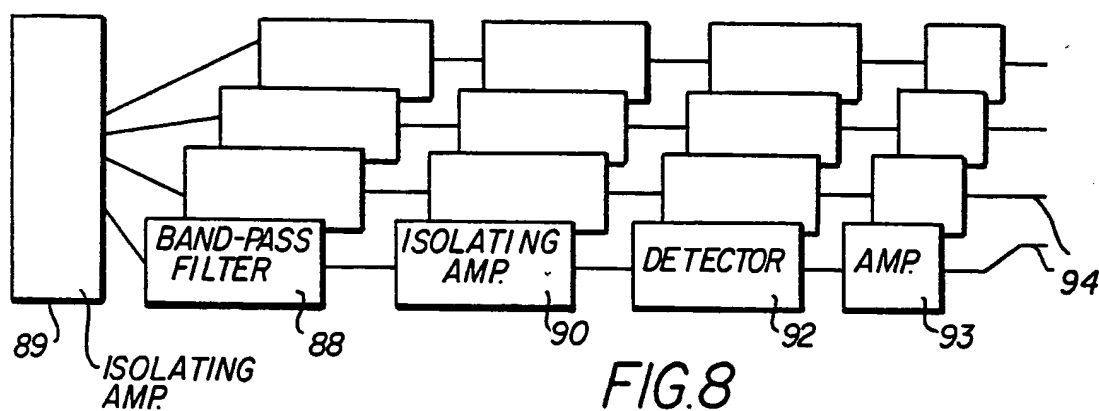
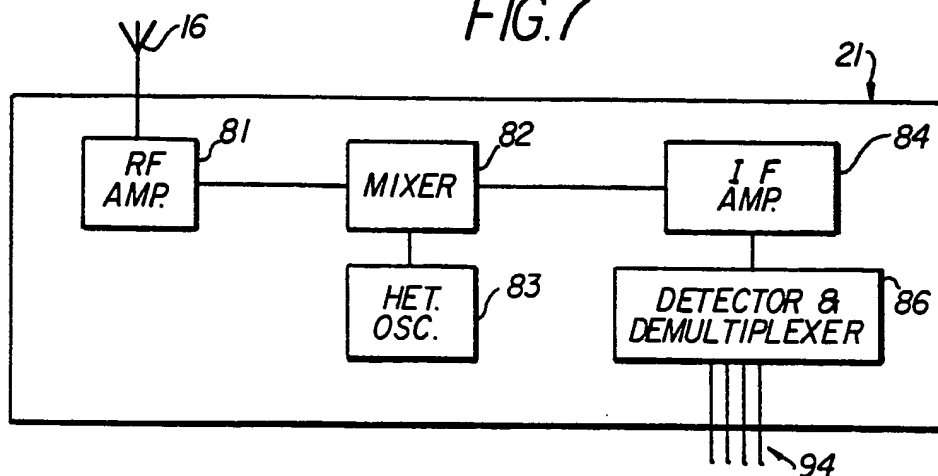


FIG. 8

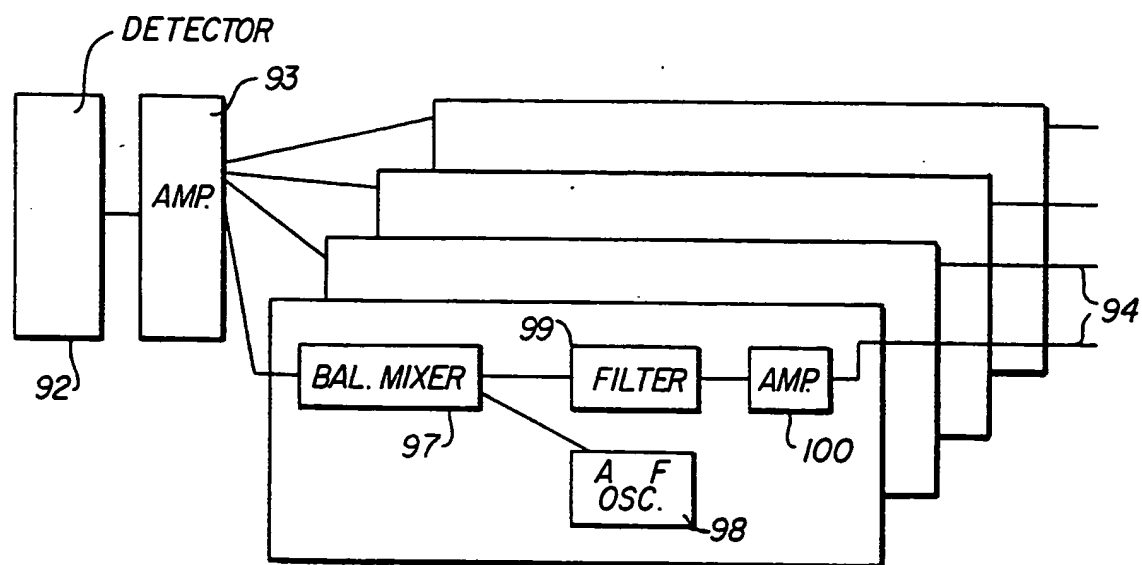
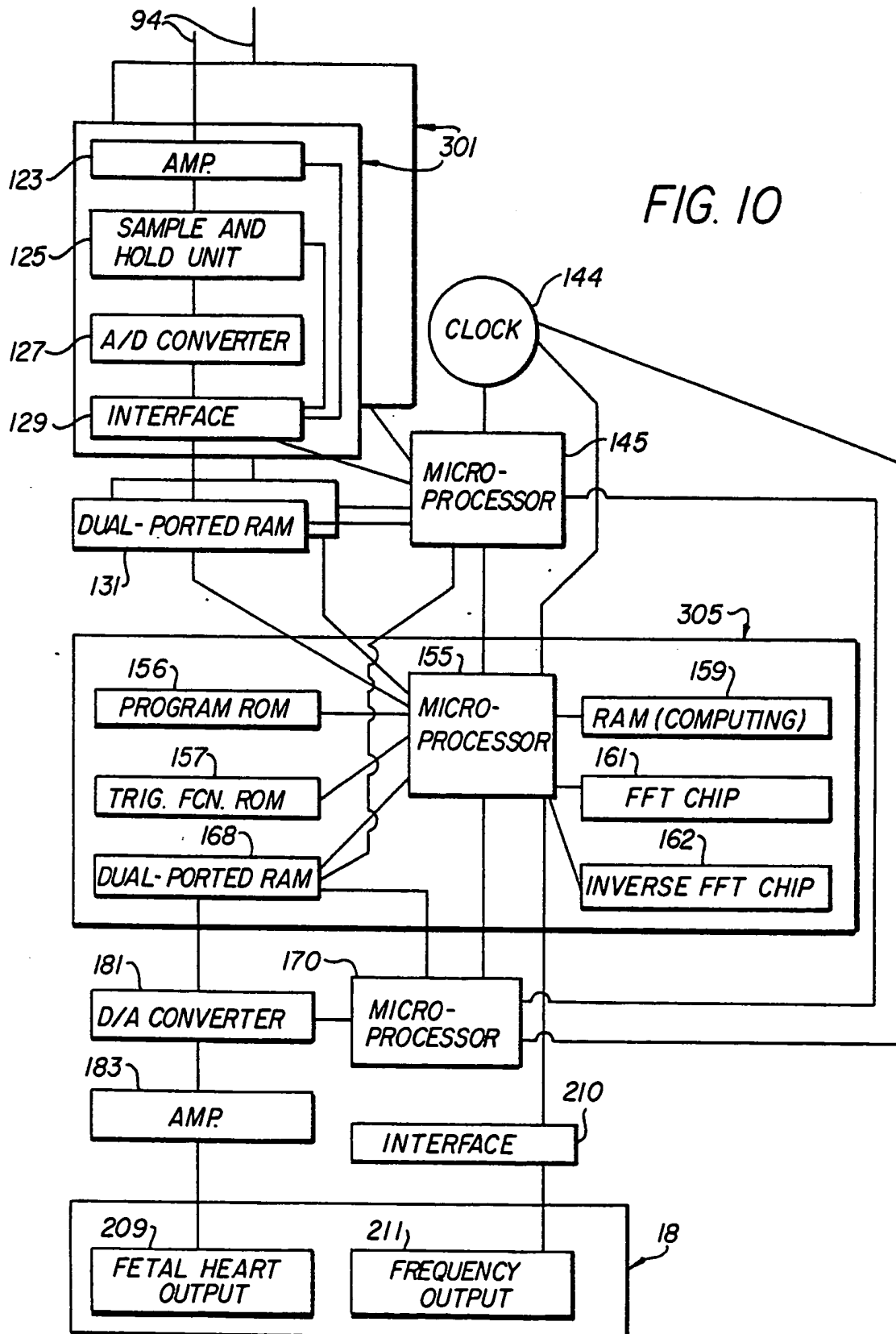


FIG. 9

7/10



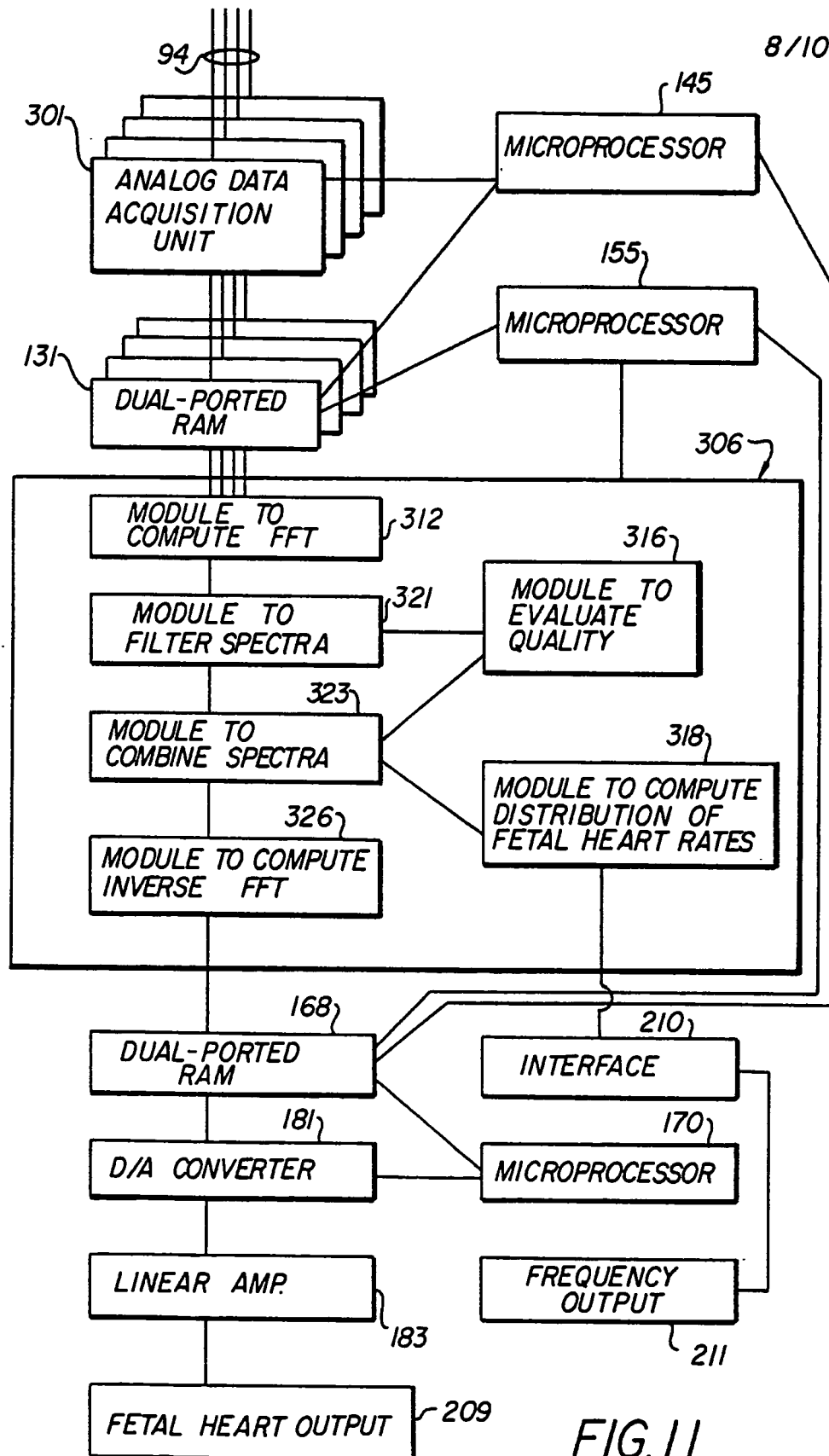


FIG. 11



9/10

FIG. 12

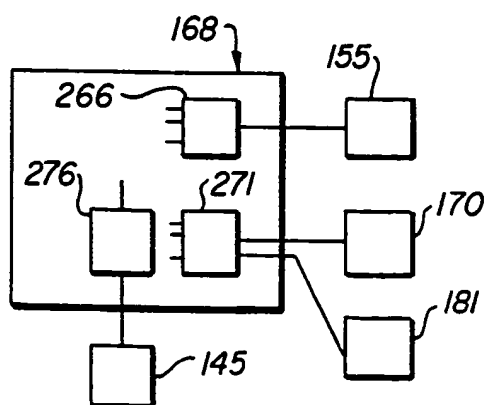
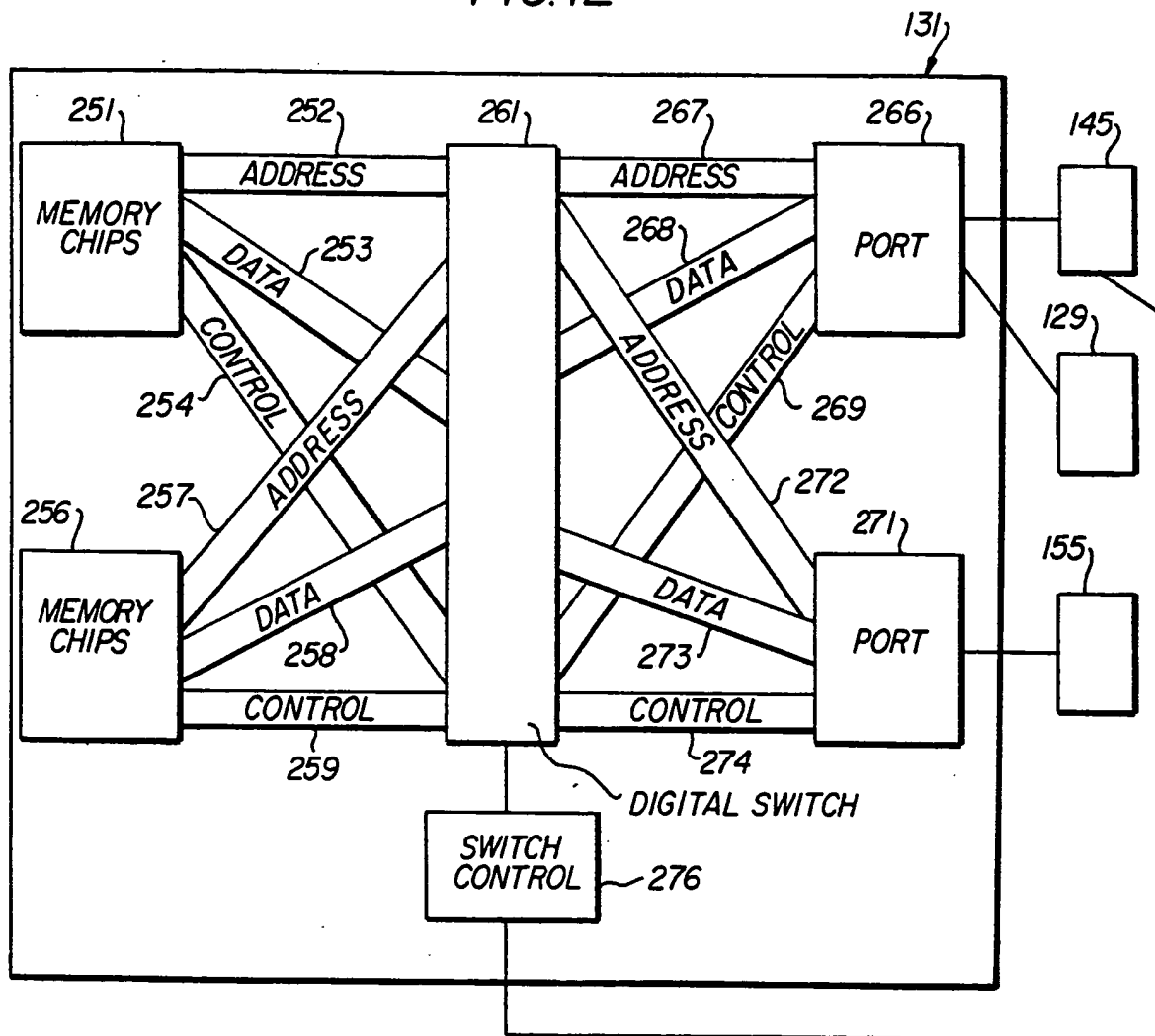


FIG. 13

10/10

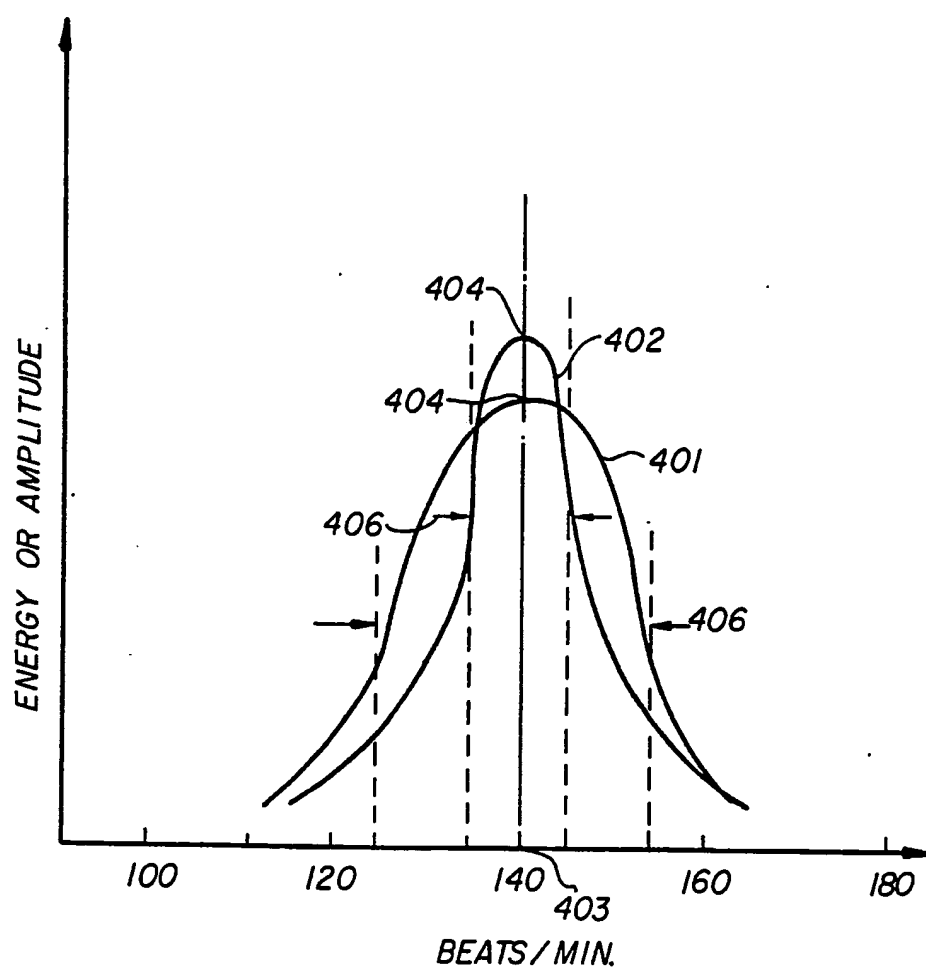


FIG. 14